

FIG. 1 (PRIOR ART)

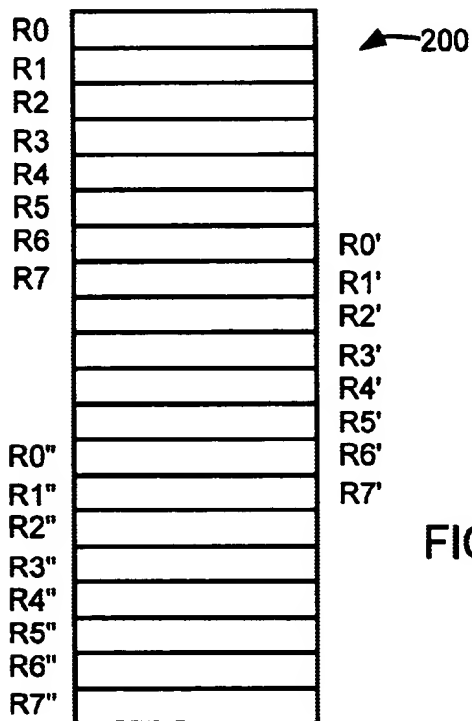


FIG. 2 (PRIOR ART)

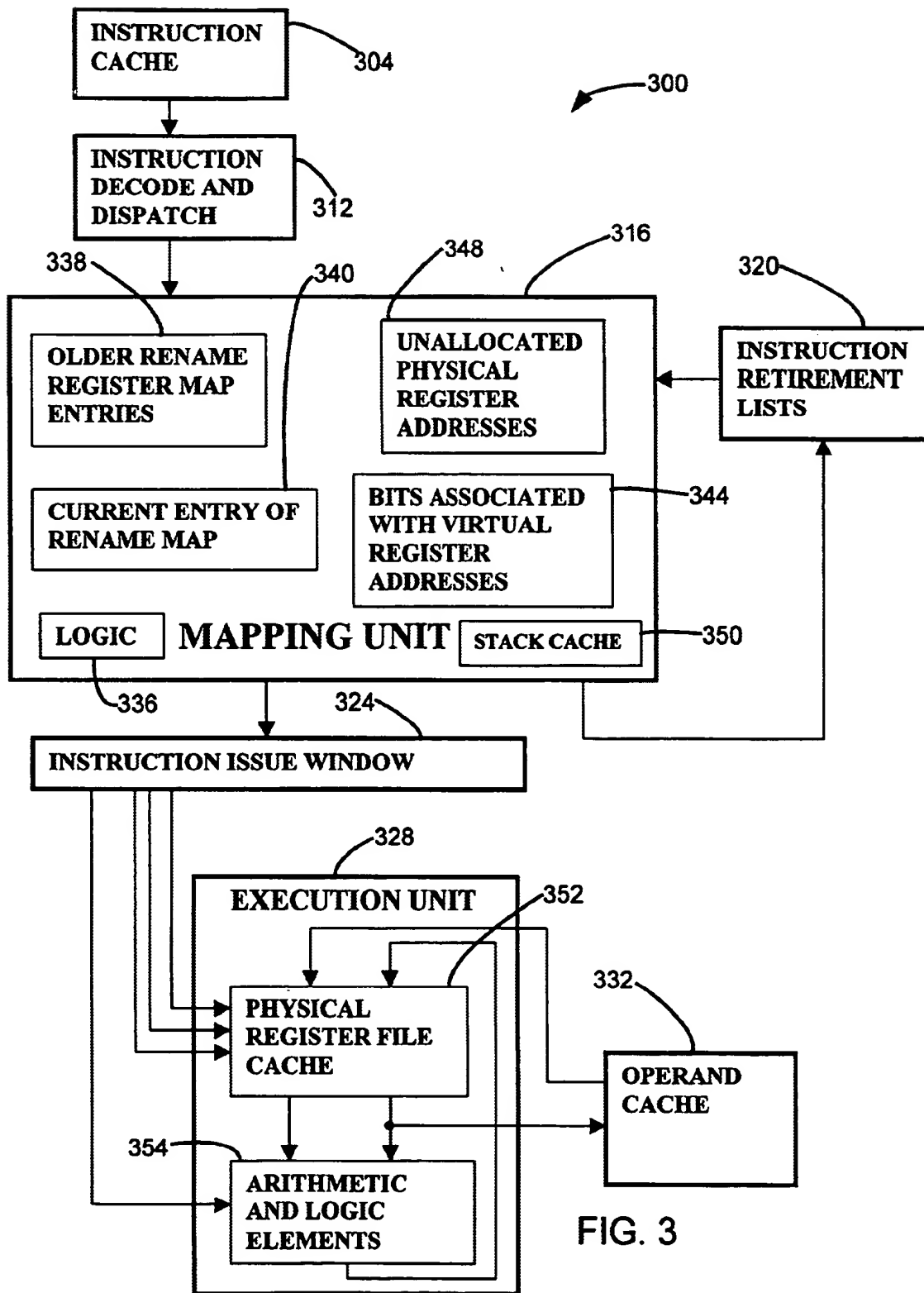


FIG. 3

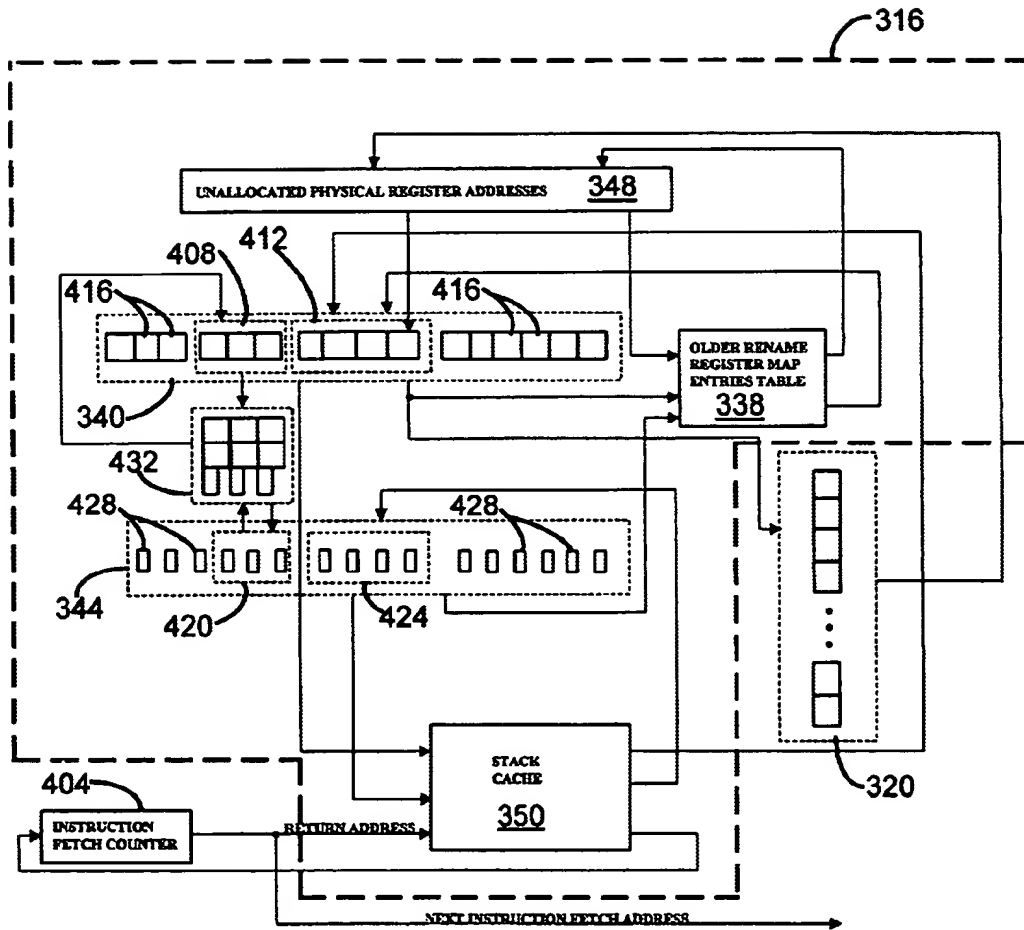


FIG. 4

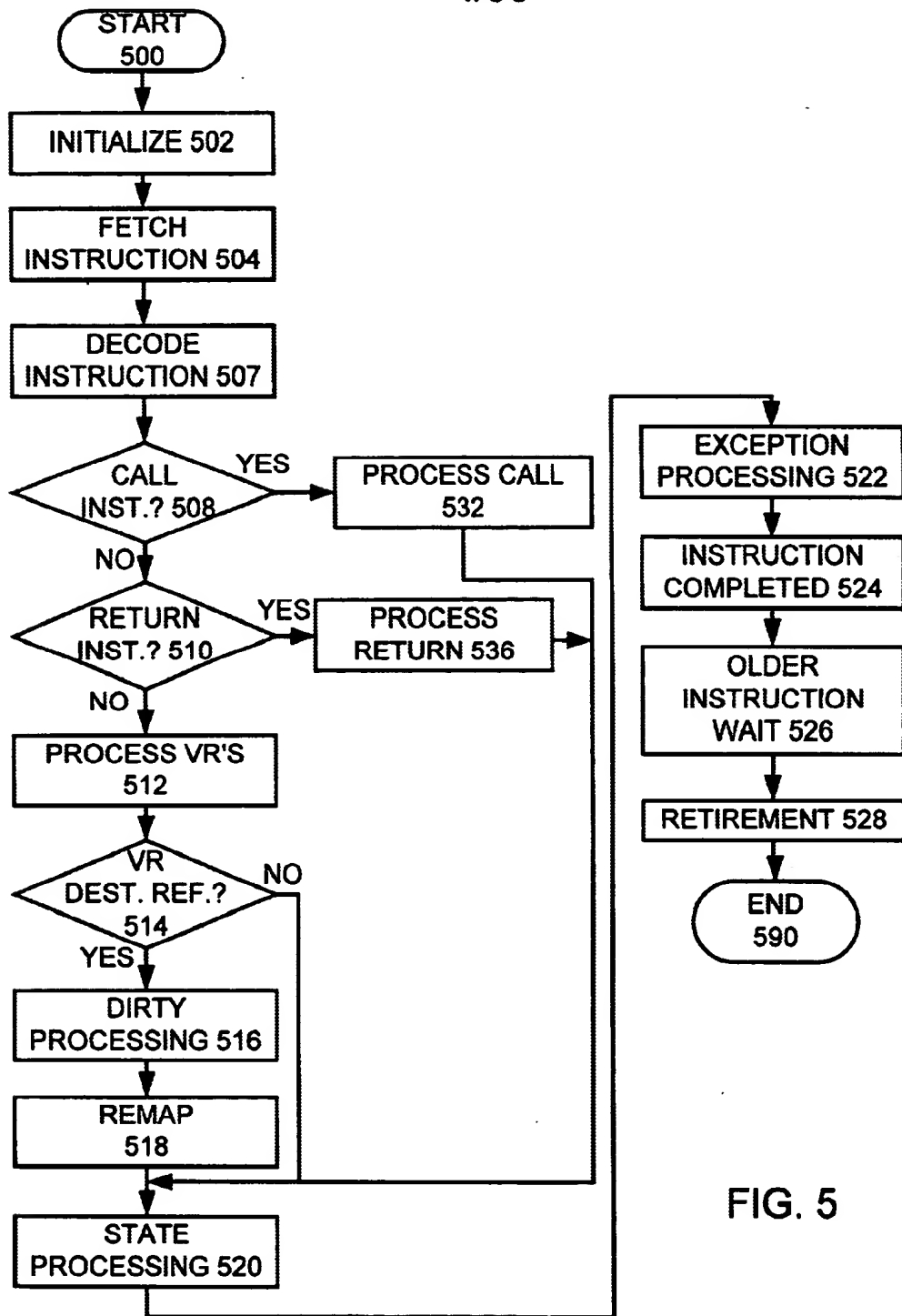
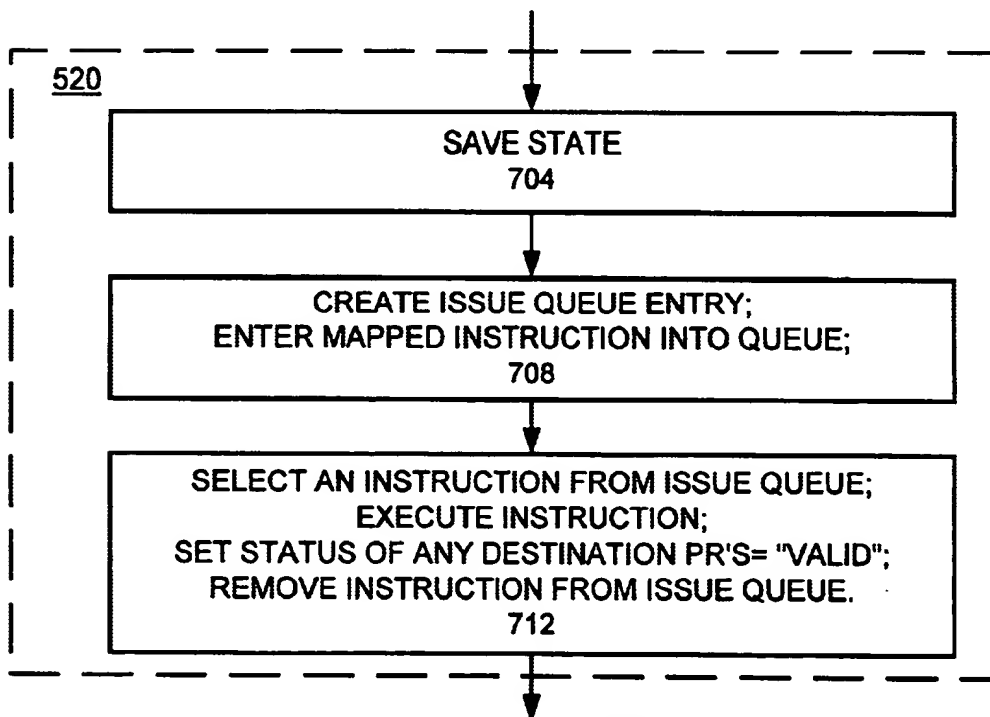
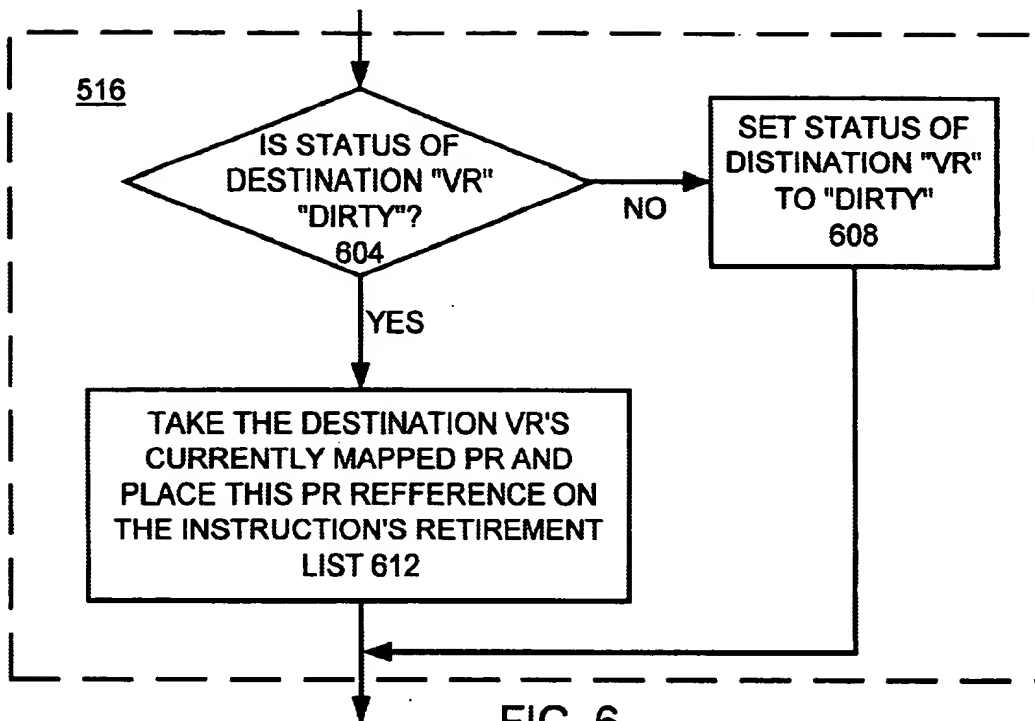


FIG. 5



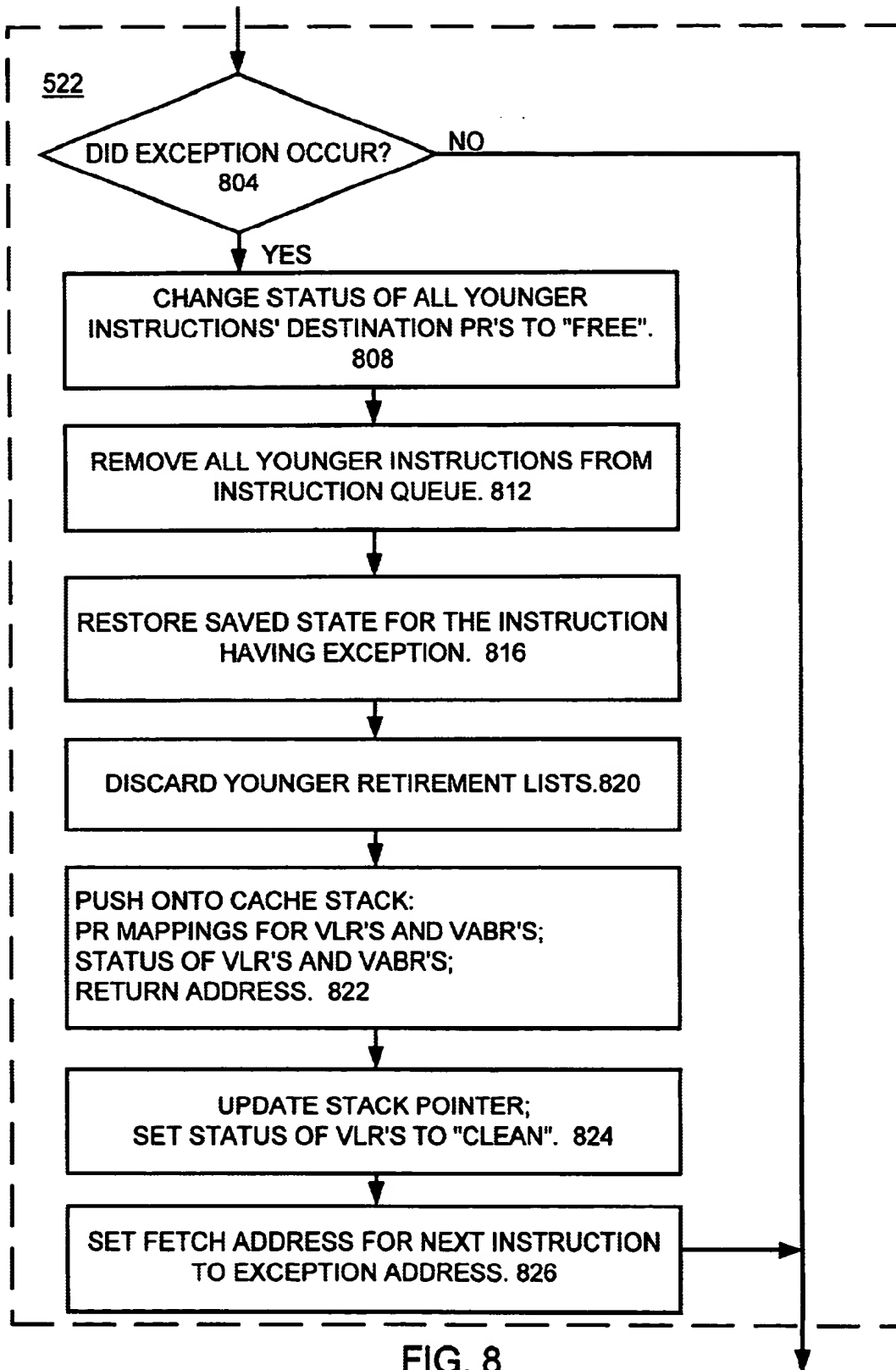


FIG. 8

7/90

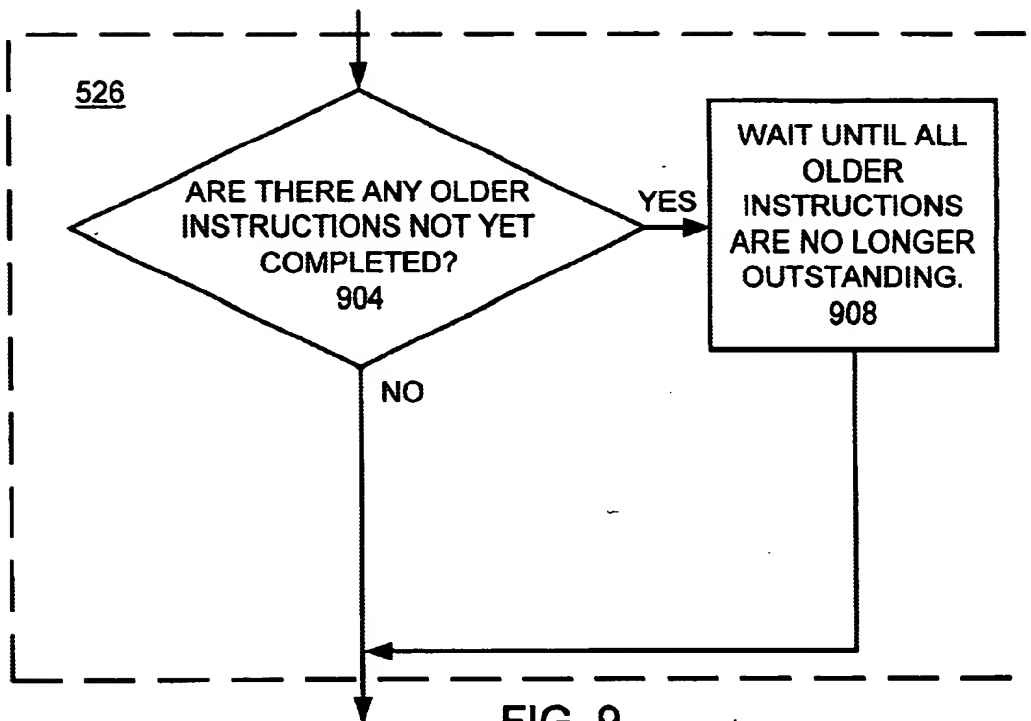


FIG. 9

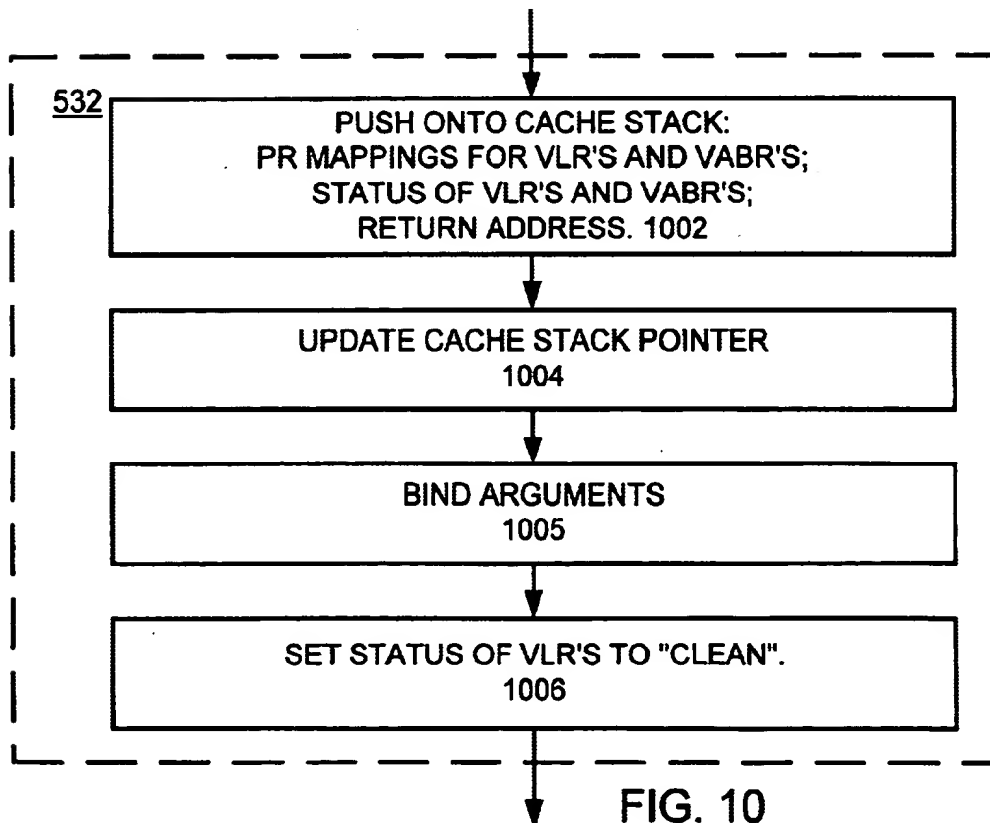


FIG. 10

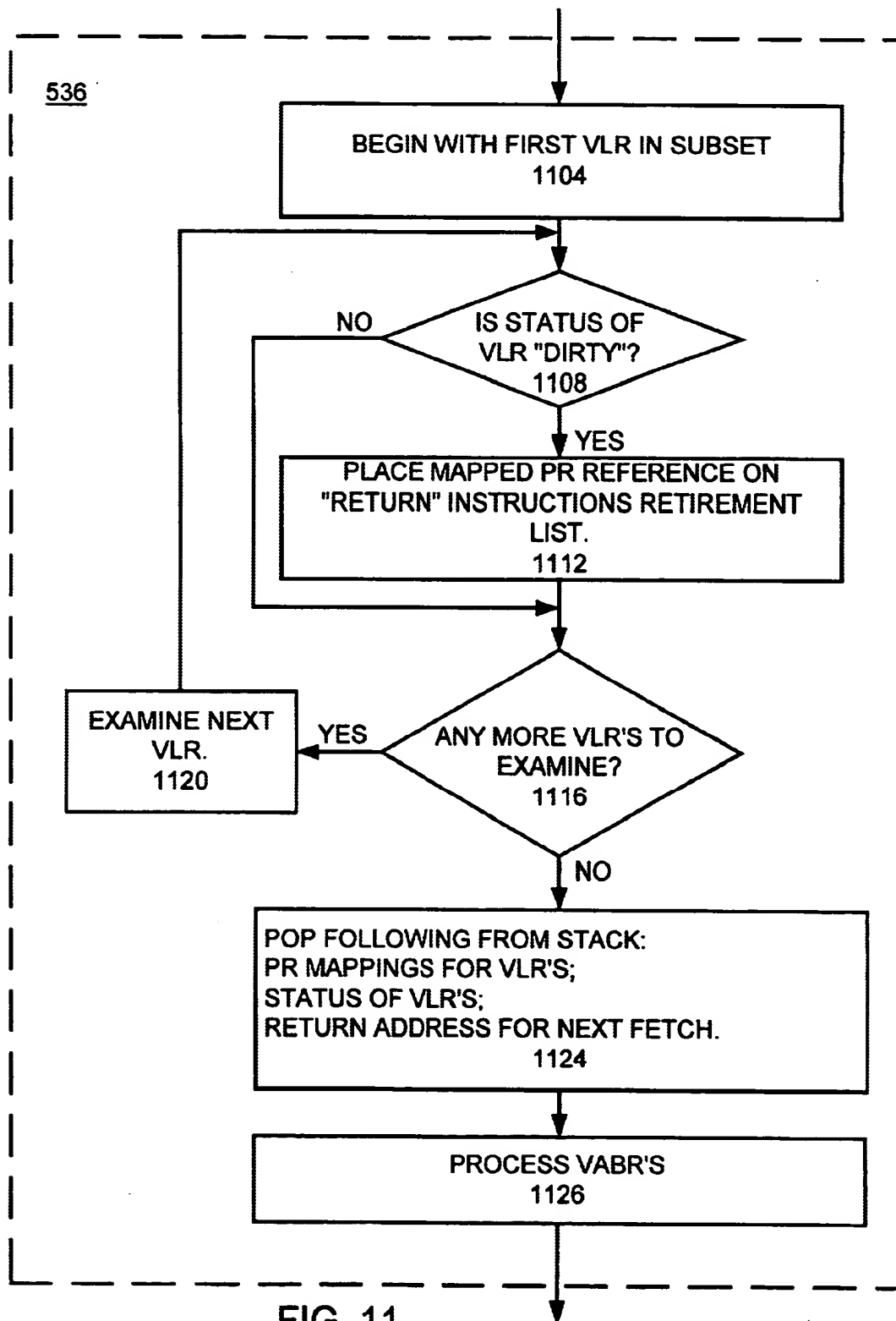


FIG. 11

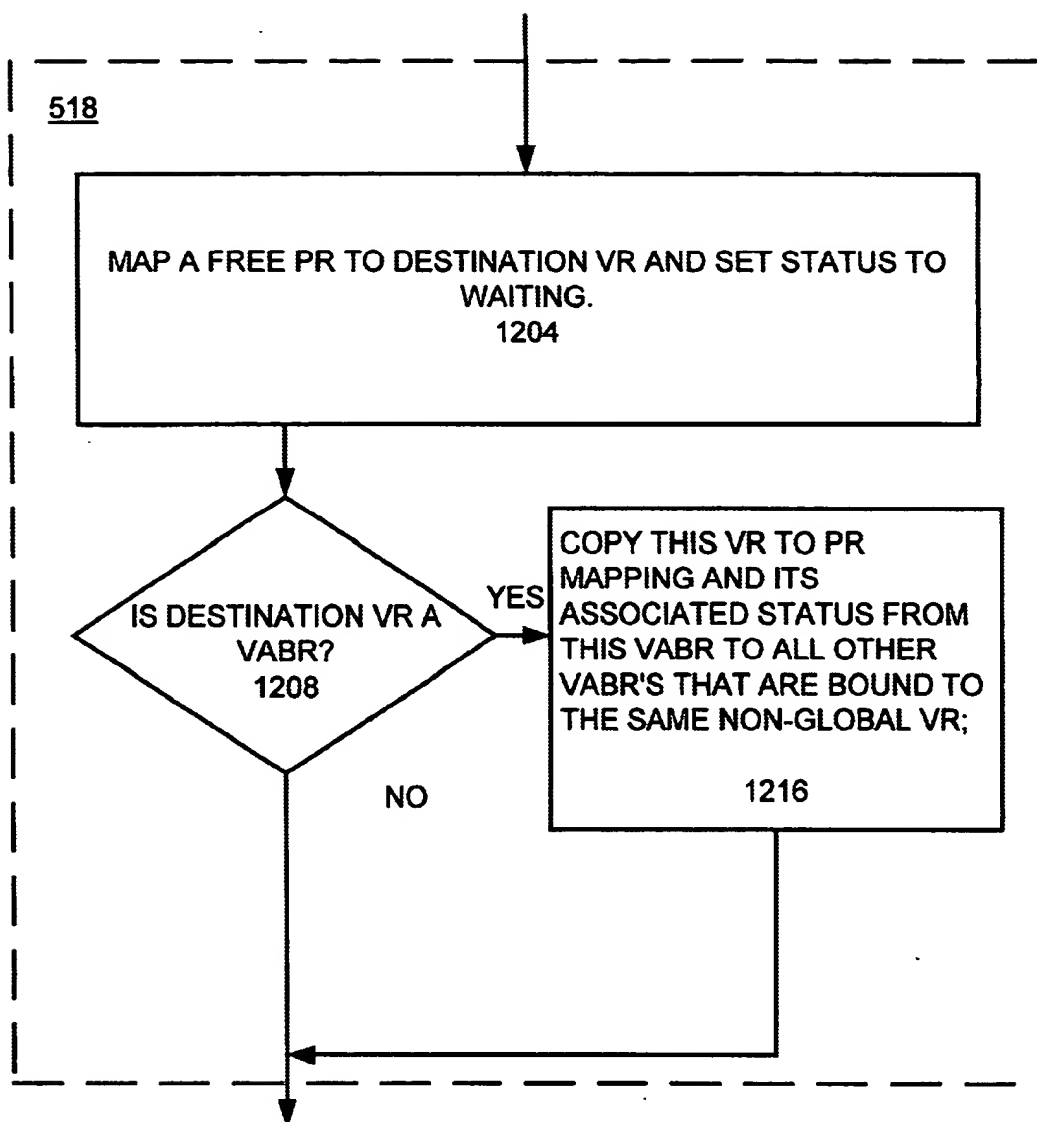


FIG. 12

10/90

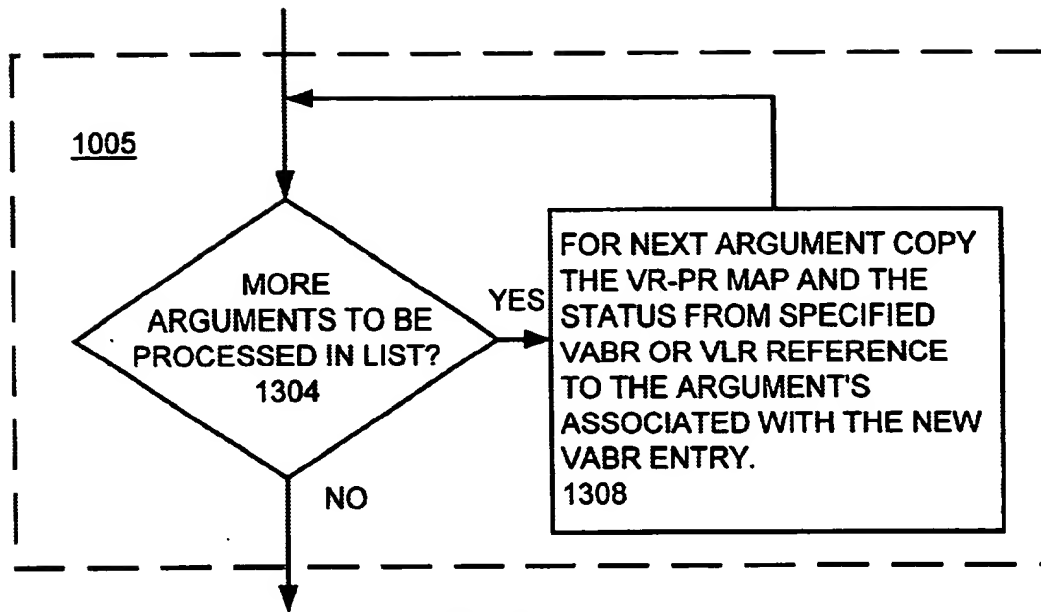


FIG. 13

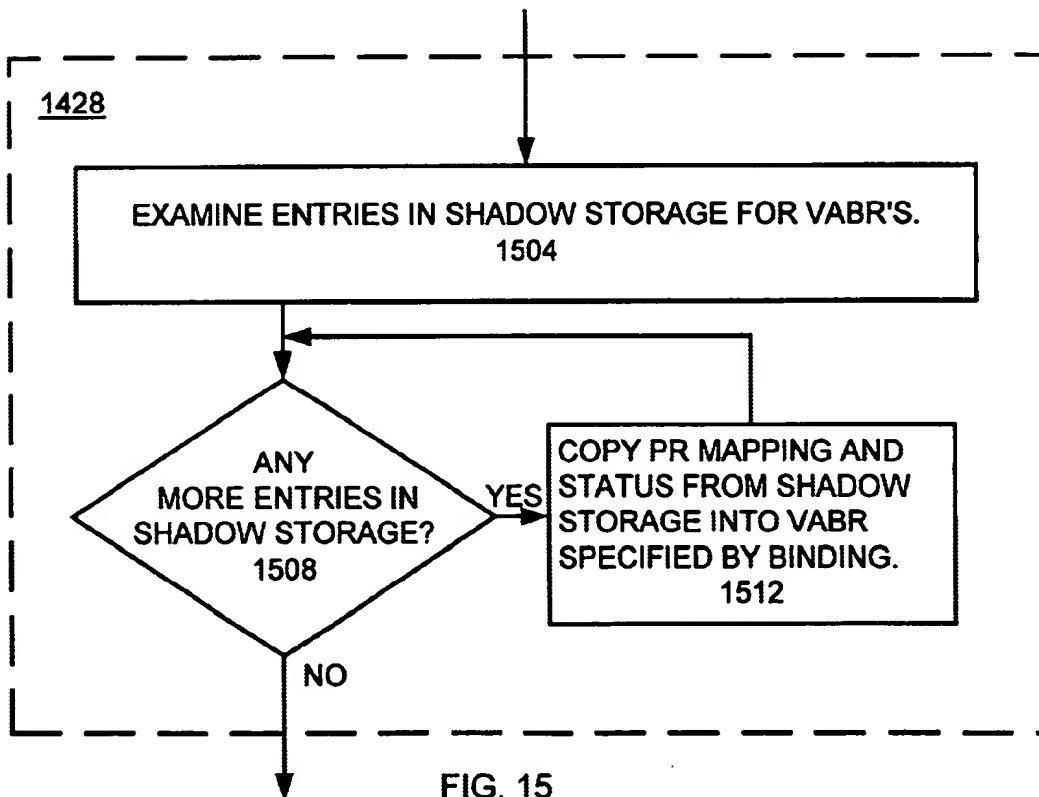


FIG. 15

FIG. 10

11/90

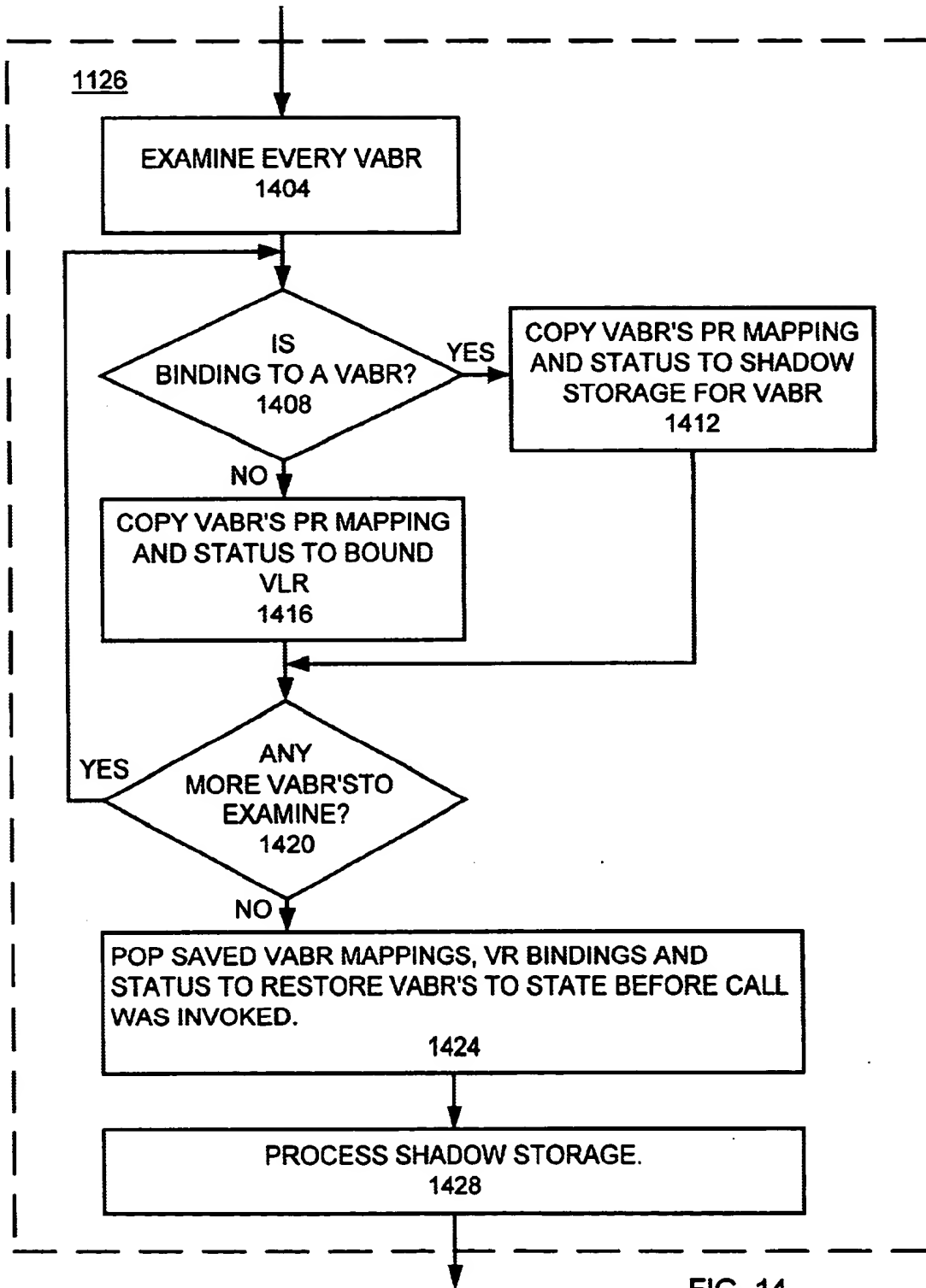


FIG. 14

03855-010301

12/90

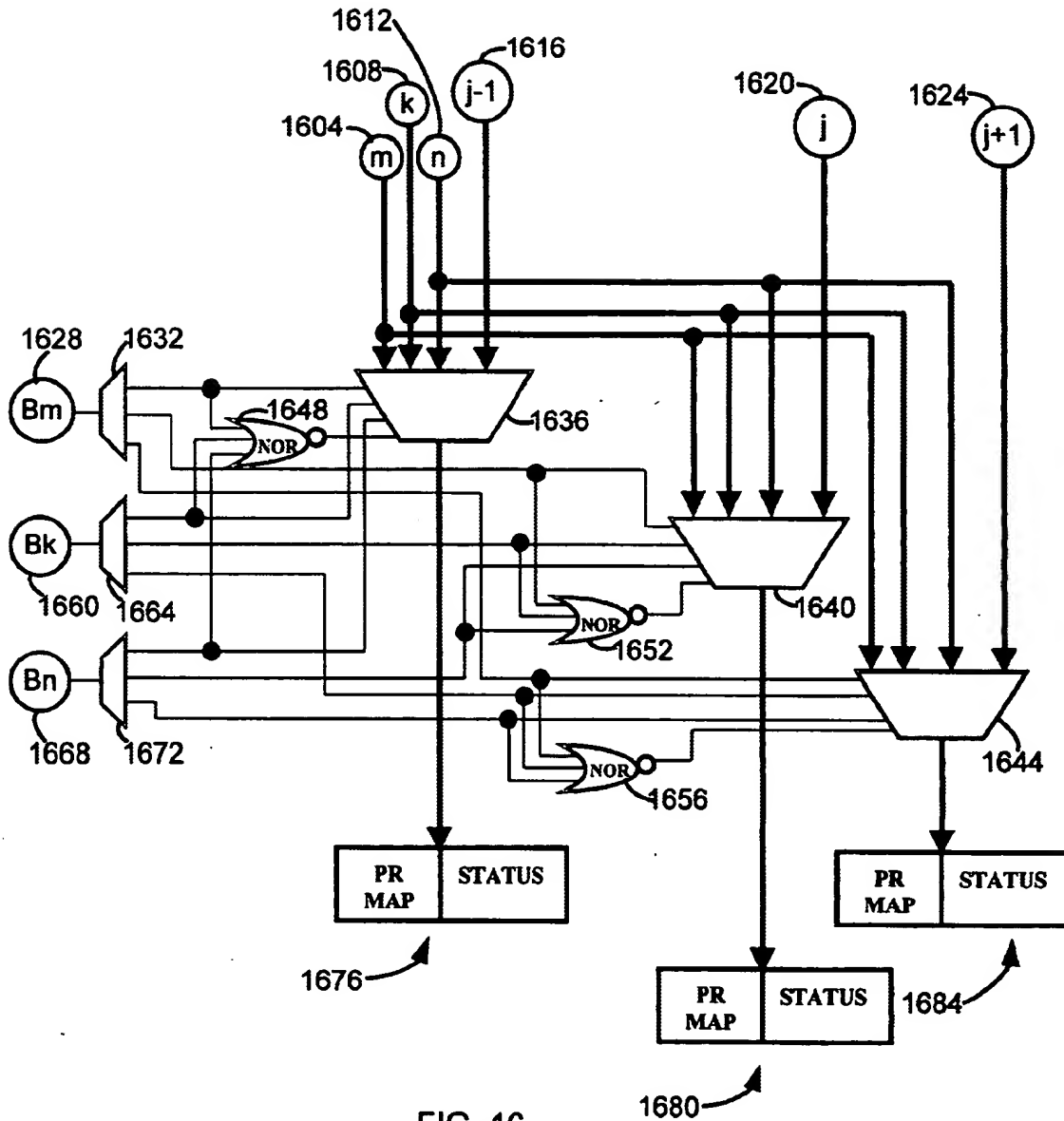


FIG. 16

FIG. 16

| FETCH | DECODE AND ISSUE | READ REGISTER FILE | EXECUTE AND WRITE RESULT BACK TO REGISTER FILE | RETIRE |
|-------|------------------------|--------------------------|---|--------|
| | | | | |

EXAMPLE PROGRAM

start of example execution

end of example execution

FIG. 18

CLOCK 1: DECODE STAGE
INITIAL PHYSICAL REGISTER STATE

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | DESCRIPTION |
|--------------------------|------|--------------|-------|------------------------|
| 0 | 0 | 1 | 3 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | EXAMPLE INITIALIZATION |
| 2 | 0 | 1 | 7 | EXAMPLE INITIALIZATION |
| 3 | 0 | 1 | 9 | EXAMPLE INITIALIZATION |
| 4 | 0 | 1 | 11 | EXAMPLE INITIALIZATION |
| 5 | 0 | 1 | 13 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | EXAMPLE INITIALIZATION |
| 7 | 0 | 1 | 17 | EXAMPLE INITIALIZATION |
| 8 | 0 | 1 | 19 | EXAMPLE INITIALIZATION |
| 9 | 0 | 1 | 21 | EXAMPLE INITIALIZATION |
| 10 | 0 | 1 | 23 | EXAMPLE INITIALIZATION |
| 11 | 0 | 1 | 25 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | EXAMPLE INITIALIZATION |
| 16 | 1 | - | - | UNALLOCATED |
| 17 | 1 | - | - | UNALLOCATED |
| 18 | 1 | - | - | UNALLOCATED |
| 19 | 1 | - | - | UNALLOCATED |
| 20 | 1 | - | - | UNALLOCATED |
| 21 | 1 | - | - | UNALLOCATED |
| 22 | 1 | - | - | UNALLOCATED |
| 23 | 1 | - | - | UNALLOCATED |
| 24 | 1 | - | - | UNALLOCATED |
| ETC. | 1 | - | - | UNALLOCATED |

FIG. 19

Initial Mapping States

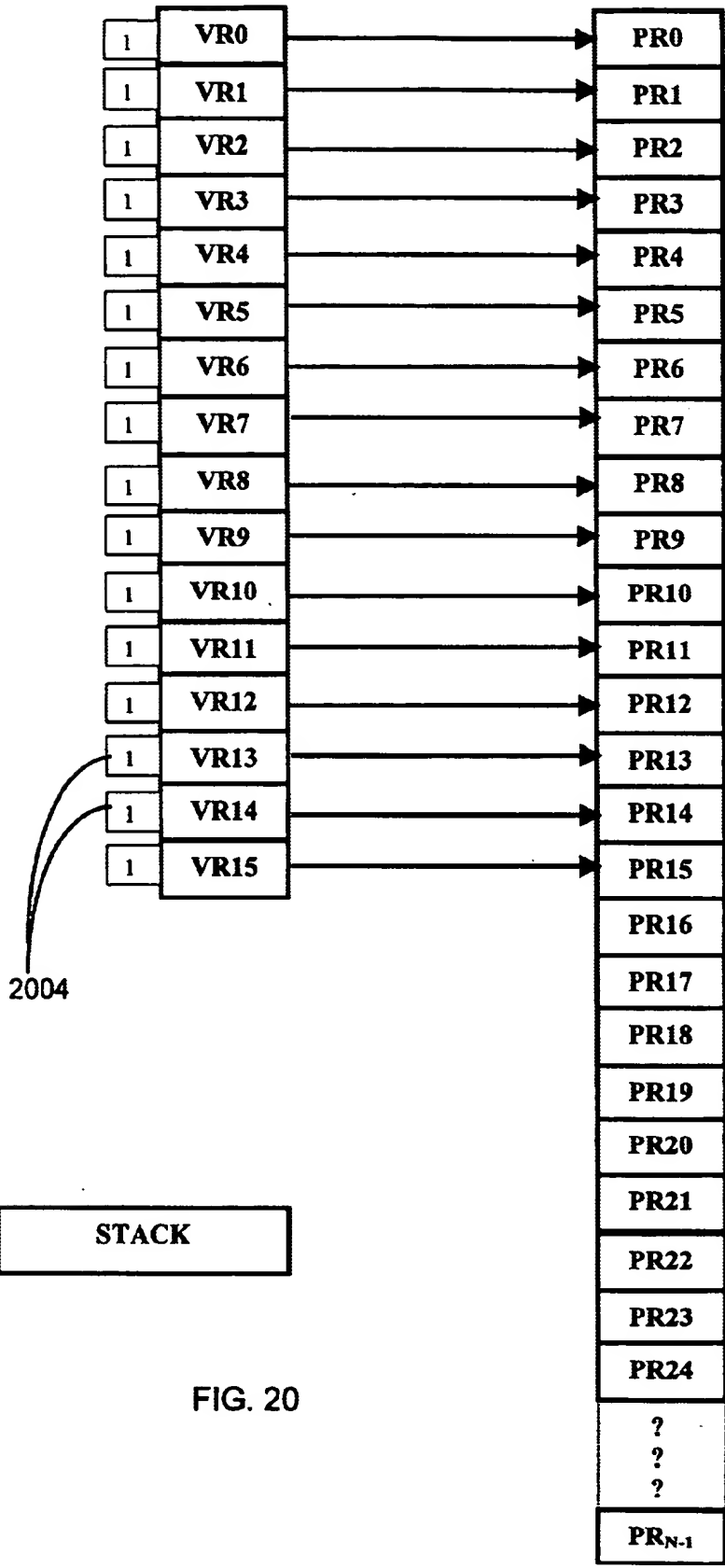


FIG. 20

16/90

| INSTRUCTION NUMBER | INSTRUCTION | DESCRIPTION | EFFECT OF INSTRUCTION |
|-----------------------|--------------------|-------------------|--|
| 1 | ADD VR0, VR2, VR4 | VR0 + VR2 → VR4 | 10 → VR4 |
| 2 | LIM VR8, #22 | 22 → VR8 | 22 ₁₀ → VR8 |
| 3 | SUB VR3, VR2, VR3 | VR3 - VR2 → VR3 | 2 → VR3 |
| 4 | ADD VR4, VR3, VR3 | VR4 + VR3 → VR3 | 12 → VR3 |
| 5 | MUL VR4, VR5, VR6 | VR4 * VR5 → VR6 | 130 → VR6 |
| 6 | CALL A | CALL subroutine A | VR6—VR9 available as scratch registers |
| 7 | ADD VR6, VR3, VR10 | VR6 + VR3 → VR10 | 142 → VR10 |
| 8 | SUB VR2, VR3, VR8 | VR2 - VR3 → VR8 | -5 → VR8 (use VR8 as scratch register) |
| 9 | MUL VR8, VR1, VR7 | VR8 * VR1 → VR7 | -25 → VR7 (use VR7 as scratch register) |
| 10 | CALL B | CALL subroutine B | VR6—VR9 available as scratch registers |
| 11 | ADD VR1, VR2, VR6 | VR1 + VR2 → VR6 | 12 → VR6 (use VR6 as scratch register) |
| 12 | ADD VR3, VR7, VR7 | VR3 + VR7 → VR7 | -13 → VR7 (use VR7 as scratch register) |
| 13 | MUL VR6, VR7, VR1 | VR6 * VR7 → VR1 | -156 → VR1 |
| 14 | RET | RETURN | restore value of 130 to VR6 and -25 to VR7 |
| 15 | ADD VR8, VR7, VR1 | VR8 + VR7 → VR2 | -30 → VR2 |
| 16 | RET | RETURN | restore value of 17 to VR7 and 22 to VR8 |
| 17 | ADD VR8, VR1, VR1 | VR8 + VR1 → VR1 | -134 → VR1 |
| 18 | ADD VR8, VR2, VR2 | VR8 + VR2 → VR2 | -8 → VR2 |

EXAMPLE INSTRUCTION FLOW

FIG. 21

| INSTRUCTION NUMBER | VIRTUAL REGISTER NUMBER: | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----------------------|--------------------------|----------------|---|------|-----|----|----|----|-----|-----|----|----|-----|----|----|----|----|----|
| | INSTRUCTION | INITIAL VALUE: | | | | | | | | | | | | | | | | |
| 1 | ADD VR0, VR2, VR4 | | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 2 | LIM VR8, #22 | | 3 | 5 | 7 | 9 | 10 | 13 | 15 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 3 | SUB VR3, VR2, VR3 | | 3 | 5 | 7 | 2 | 10 | 13 | 15 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 4 | ADD VR4, VR3, VR3 | | 3 | 5 | 7 | 12 | 10 | 13 | 15 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 5 | MUL VR4, VR5, VR6 | | 3 | 5 | 7 | 12 | 10 | 13 | 130 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 6 | CALL A | | 3 | 5 | 7 | 12 | 10 | 13 | 130 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 7 | ADD VR6, VR3, VR10 | | 3 | 5 | 7 | 12 | 10 | 13 | 130 | 17 | 22 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 8 | SUB VR2, VR3, VR8 | | 3 | 5 | 7 | 12 | 10 | 13 | 130 | 17 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 9 | MUL VR8, VR1, VR7 | | 3 | 5 | 7 | 12 | 10 | 13 | 130 | -25 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 10 | CALL B | | 3 | 5 | 7 | 12 | 10 | 13 | 130 | -25 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 11 | ADD VR1, VR2, VR6 | | 3 | 5 | 7 | 12 | 10 | 13 | 12 | -25 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 12 | ADD VR3, VR7, VR7 | | 3 | 5 | 7 | 12 | 10 | 13 | 12 | -13 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 13 | MUL VR6, VR7, VR1 | | 3 | -156 | 7 | 12 | 10 | 13 | 12 | -13 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 14 | RET | | 3 | -156 | 7 | 12 | 10 | 13 | 130 | -25 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 15 | ADD VR8, VR7, VR1 | | 3 | -156 | -30 | 12 | 10 | 13 | 130 | -25 | -5 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 16 | RET | | 3 | -156 | -30 | 12 | 10 | 13 | 130 | 17 | 22 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 17 | ADD VR8, VR1, VR1 | | 3 | -134 | -30 | 12 | 10 | 13 | 130 | 17 | 22 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |
| 18 | ADD VR8, VR2, VR2 | | 3 | -134 | -8 | 12 | 10 | 13 | 130 | 17 | 22 | 21 | 142 | 25 | 27 | 29 | 31 | 33 |

CONTENTS OF VIRTUAL REGISTERS AS INSTRUCTIONS EXECUTE

FIG. 22

| | | | |
|----------------|----------------------|-----------------------|--|
| <u>Clock 1</u> | Fetch instr. 1, 2. | | |
| <u>Clock 2</u> | Fetch instr. 3, 4; | Decode instr. 1, 2. | |
| <u>Clock 3</u> | Fetch instr. 5, 6; | Decode instr. 3, 4; | Read regs. PR0, PR2 for instr. 1. |
| <u>Clock 4</u> | Fetch instr. 7, 8; | Decode instr. 5, 6; | Read regs. PR2, PR3 for instr. 3; respectively. Execute instr. 6 (CALL A). |
| <u>Clock 5</u> | Fetch instr. 9, 10; | Decode instr. 7, 8; | Read regs. PR5, PR16 for instr. 5; |
| <u>Clock 6</u> | Fetch instr. 11, 12; | Decode instr. 9, 10; | Read regs. PR16, PR18 for instr. 4; |
| <u>Clock 7</u> | Fetch instr. 13, 14; | Decode instr. 11, 12; | Execute instr. 5 and store result in PR20; Execute instr. 10 (CALL B); Retire instr. 3. |
| <u>Clock 8</u> | Fetch instr. 15, 16; | Decode instr. 13, 14; | Execute instr. 4 and store result in PR19. |
| <u>Clock 9</u> | Fetch instr. 17, 18; | Decode instr. 15, 16; | Execute instr. 14(Return). Retire instr. 4, 5, 6. |
| | | | Execute instr. 7 and 8 and store results in PR21 and PR22 respectively. Execute instr. 16(Return). |

Clock by Clock Pipeline Description

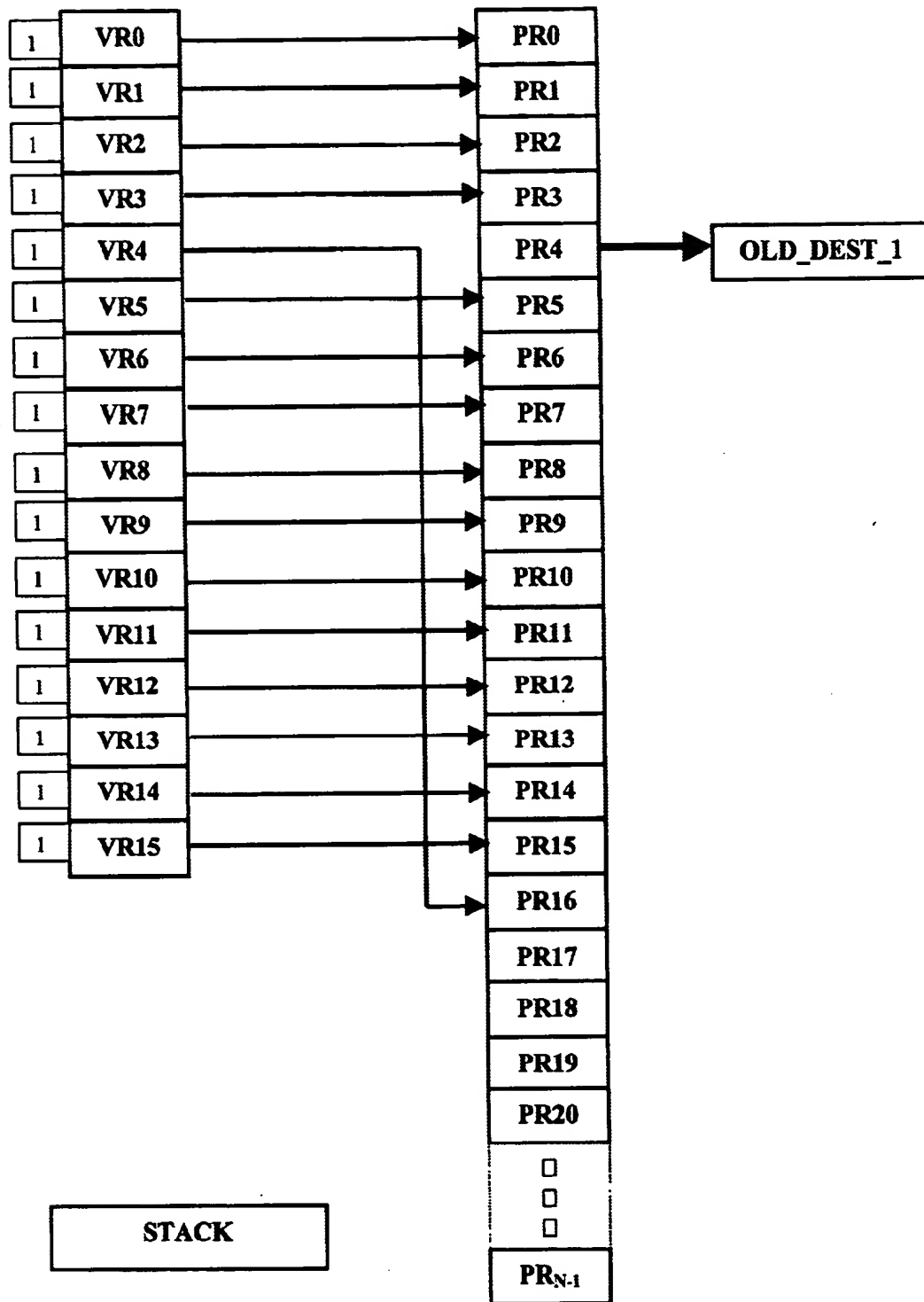
FIG. 23A

19/90

| | | |
|-----------------|--|---|
| <u>Clock 10</u> | Decode instr. 17, 18; | Execute instr. 9 and 11 and store results in PR4 and PR8 respectively. |
| <u>Clock 11</u> | | Retire instr. 9, 10, 11. |
| <u>Clock 12</u> | Read regs. PR4, PR19, PR22 for instr. 12 and 15; | |
| | Read regs. PR2, PR6, PR17 for instr. 17 and 18; | Execute instr. 12, 15 and store results in PR3 and PR6 respectively. |
| <u>Clock 13</u> | Read regs. PR3, PR8 for instr. 13; | Execute instr. 17, 18 and store results in PR18 and PR24 respectively; Retire instr. 12. |
| <u>Clock 14</u> | | Execute instr. 13 and store results in PR23. |
| <u>Clock 15</u> | | Retire instr. 13, 14, 15, 16, 17, 18. |

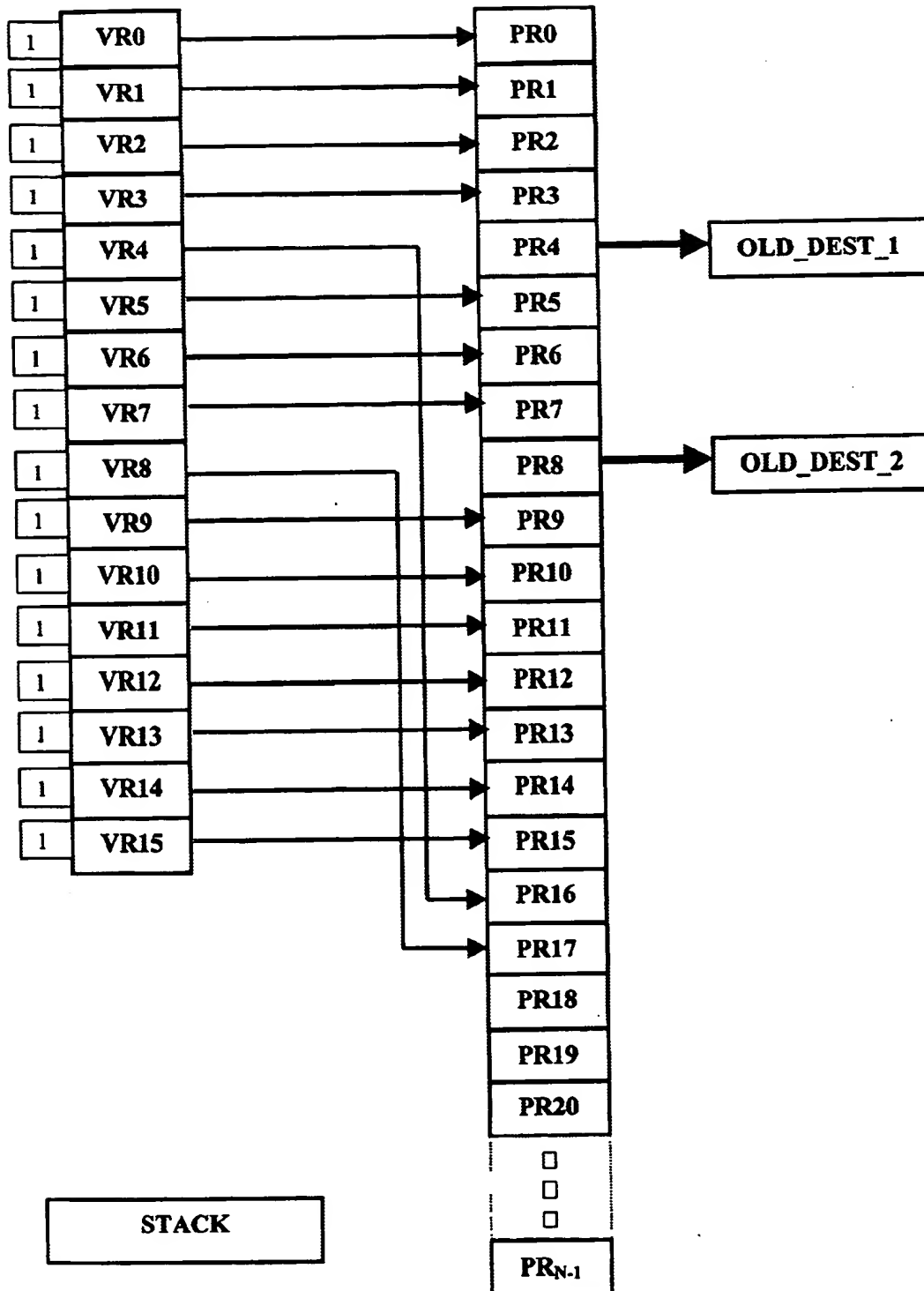
Clock by Clock Pipeline Description

FIG. 23B



INSTR. 1: ADD VR0, VR2, VR4 maps to PR0 + PR2 → PR16,
PR4 → OLD_DEST_1

FIG. 24



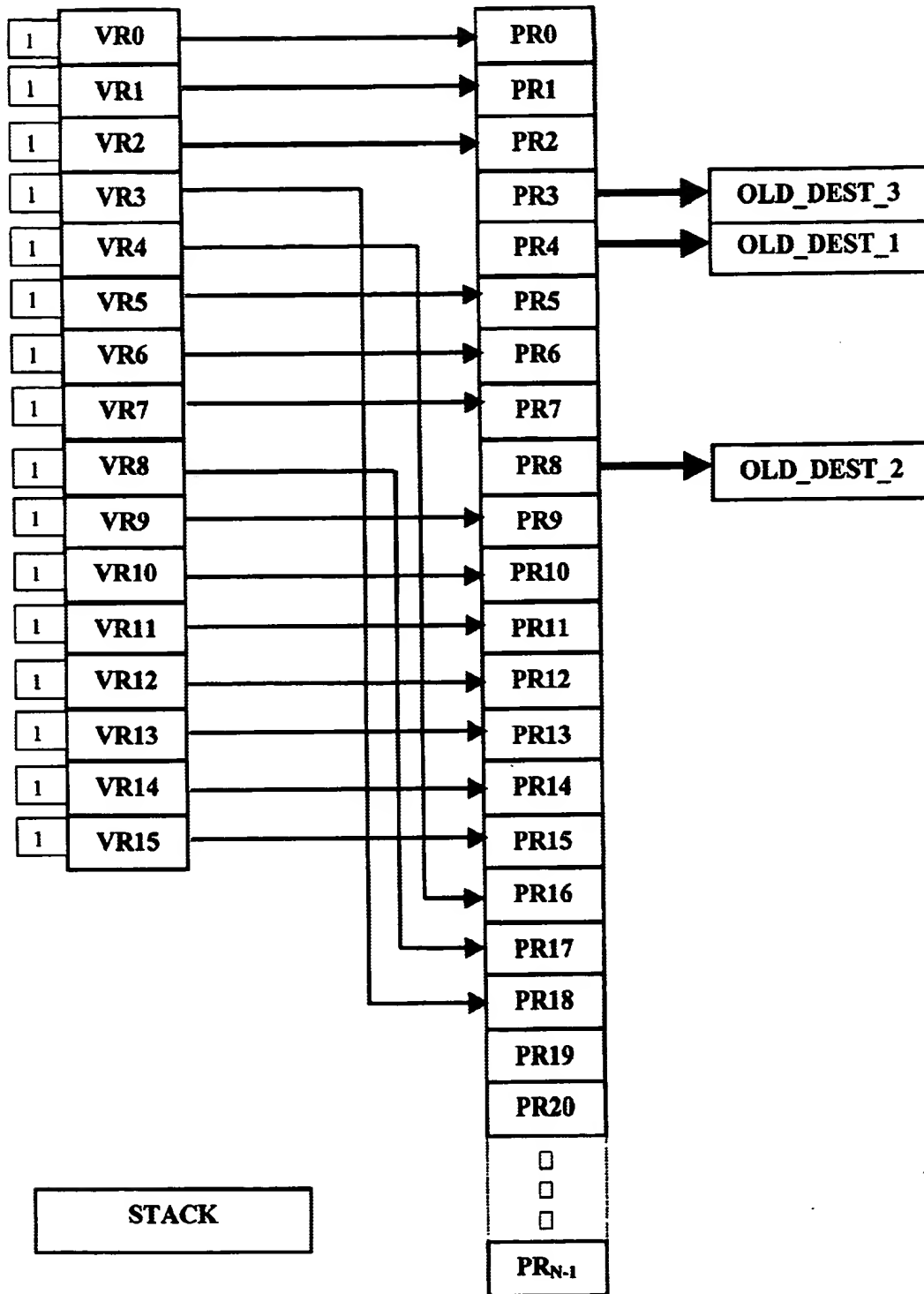
INSTR. 2: LIM VR8, #22 maps to LIM PR17, #22, PR8 → OLD_DEST_2

FIG. 25

CLOCK 2: DECODE STAGE

INSTRUCTIONS 1 & 2 PHYSICAL REGISTER STATE

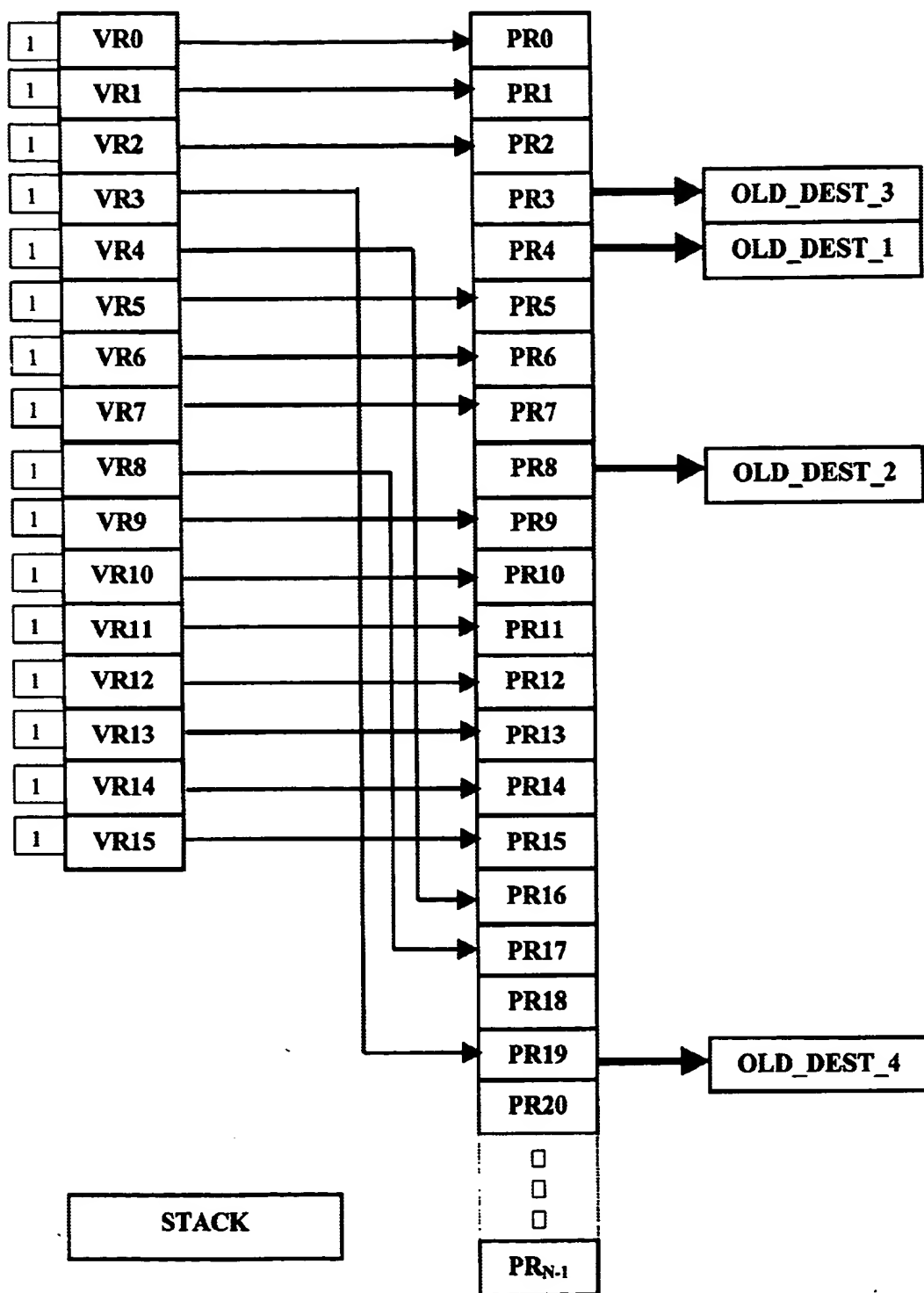
FIG. 26



INSTR. 3: SUB VR3, VR2, VR3 maps to SUB PR3, PR2, PR18,
PR3 → OLD_DEST_3

FIG. 27

24/90



INSTR. 4: ADD VR4, VR3, VR3 maps to ADD PR16, PR18, PR19,
PR18 → OLD_DEST_4

FIG. 28

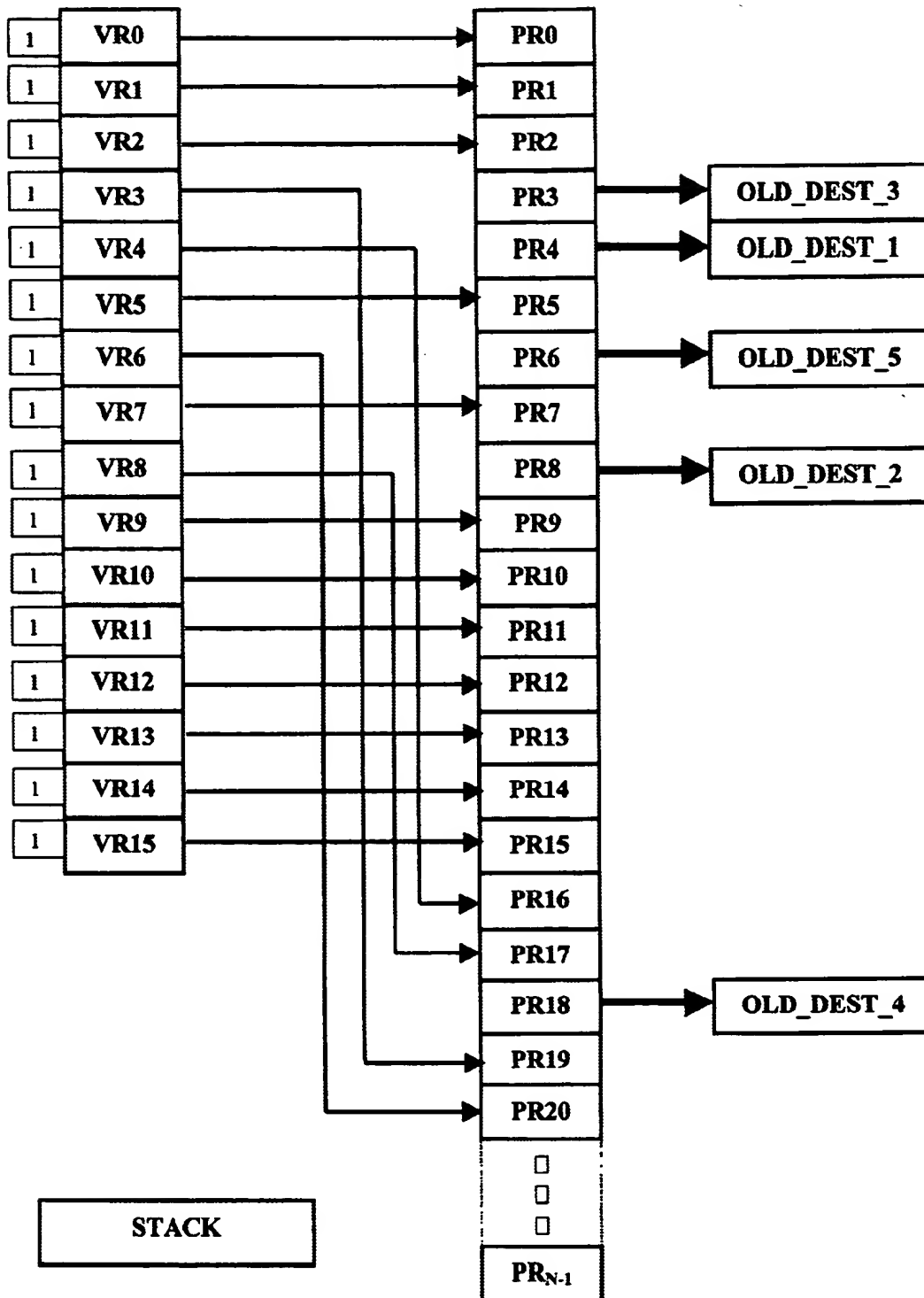
25/90

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|--------------------------------------|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | 1 | EXAMPLE INITIALIZATION |
| 2 | 0 | 1 | 7 | 2 | EXAMPLE INITIALIZATION |
| 3 | 0 | 1 | 9 | - | WAITING FOR INSTRUCTION 3 TO RETIRE |
| 4 | 0 | 1 | 11 | - | WAITING FOR INSTRUCTION 1 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | 6 | EXAMPLE INITIALIZATION |
| 7 | 0 | 1 | 17 | 7 | EXAMPLE INITIALIZATION |
| 8 | 0 | 1 | 19 | - | WAITING FOR INSTRUCTION 2 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 0 | 1 | 23 | 10 | EXAMPLE INITIALIZATION |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 0 | - | 4 | WAITING FOR INSTRUCTION 1 TO EXECUTE |
| 17 | 0 | 0 | - | 8 | WAITING FOR INSTRUCTION 2 TO EXECUTE |
| 18 | 1 | - | - | - | WAITING FOR 3 TO EXECUTE & RETIRE |
| 19 | 1 | - | - | 3 | WAITING FOR INSTRUCTION 4 TO EXECUTE |
| 20 | 1 | - | - | - | UNALLOCATED |
| 21 | 1 | - | - | - | UNALLOCATED |
| 22 | 1 | - | - | - | UNALLOCATED |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 3: DECODE STAGE
INSTRUCTIONS 3 & 4 PHYSICAL REGISTER STATE

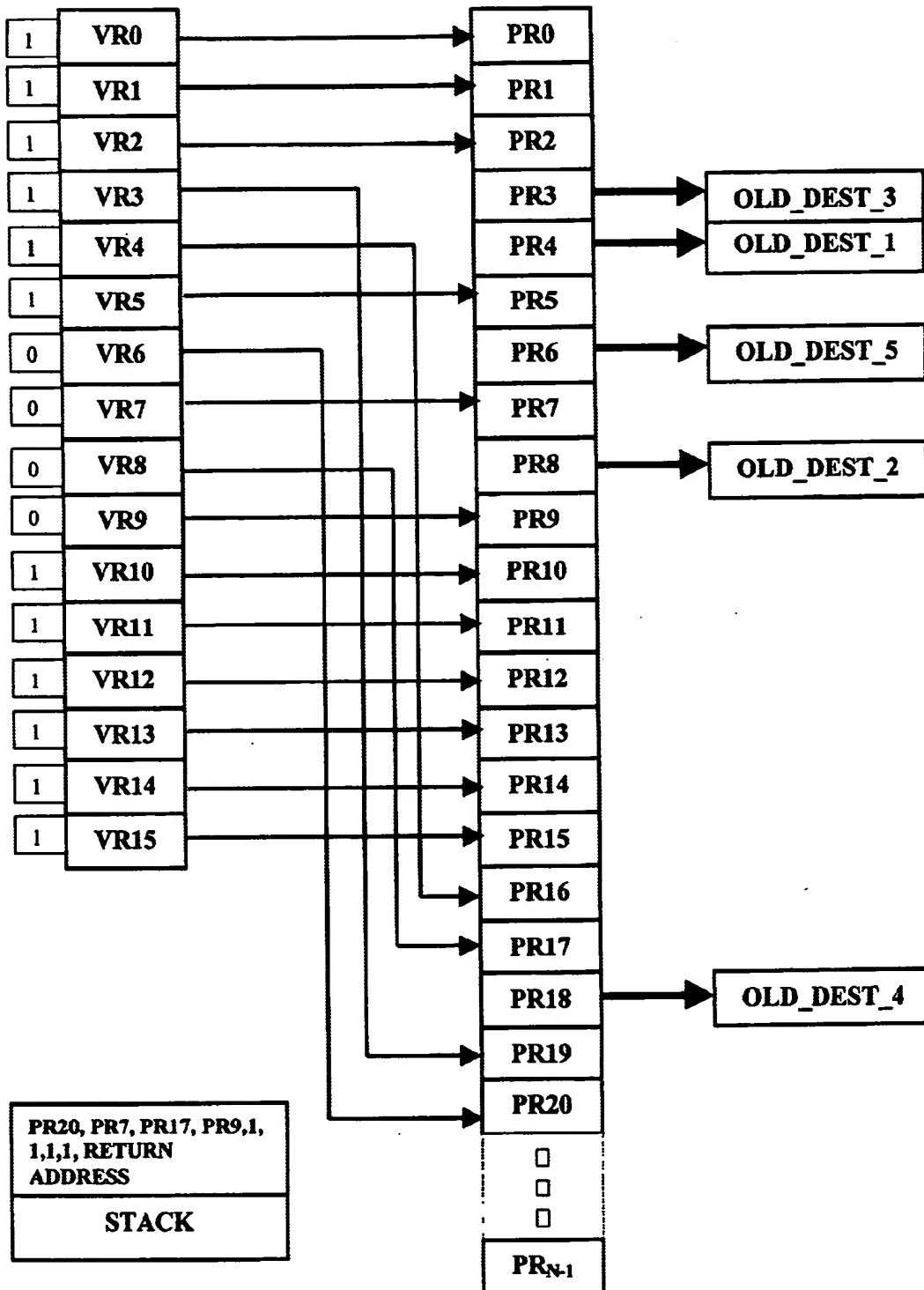
FIG. 29

TOP SECRET



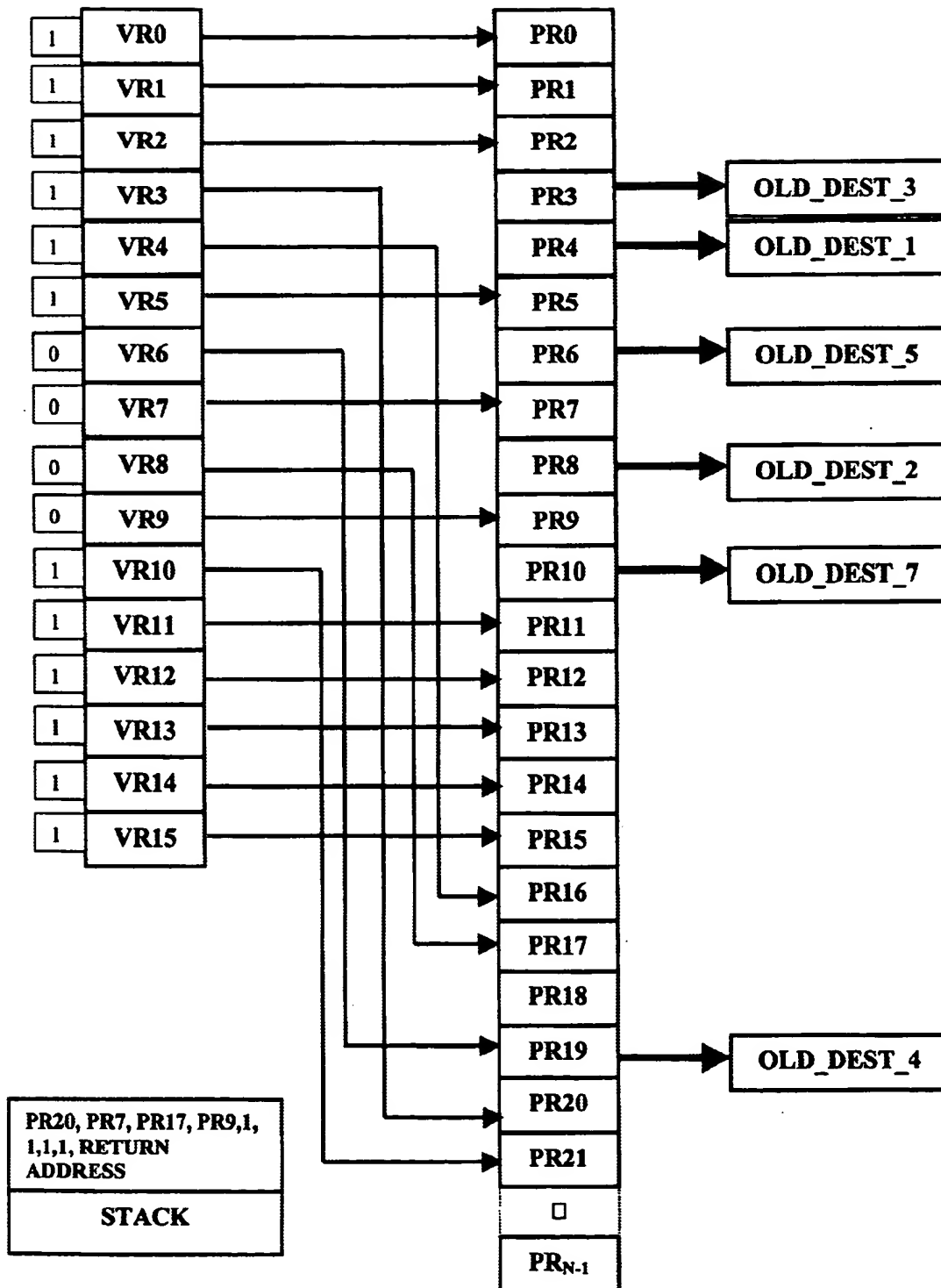
INSTR. 5: MUL VR4, VR5, VR6 maps to MUL PR16, PR5, PR20,
PR6 → OLD_DEST_5

FIG. 30



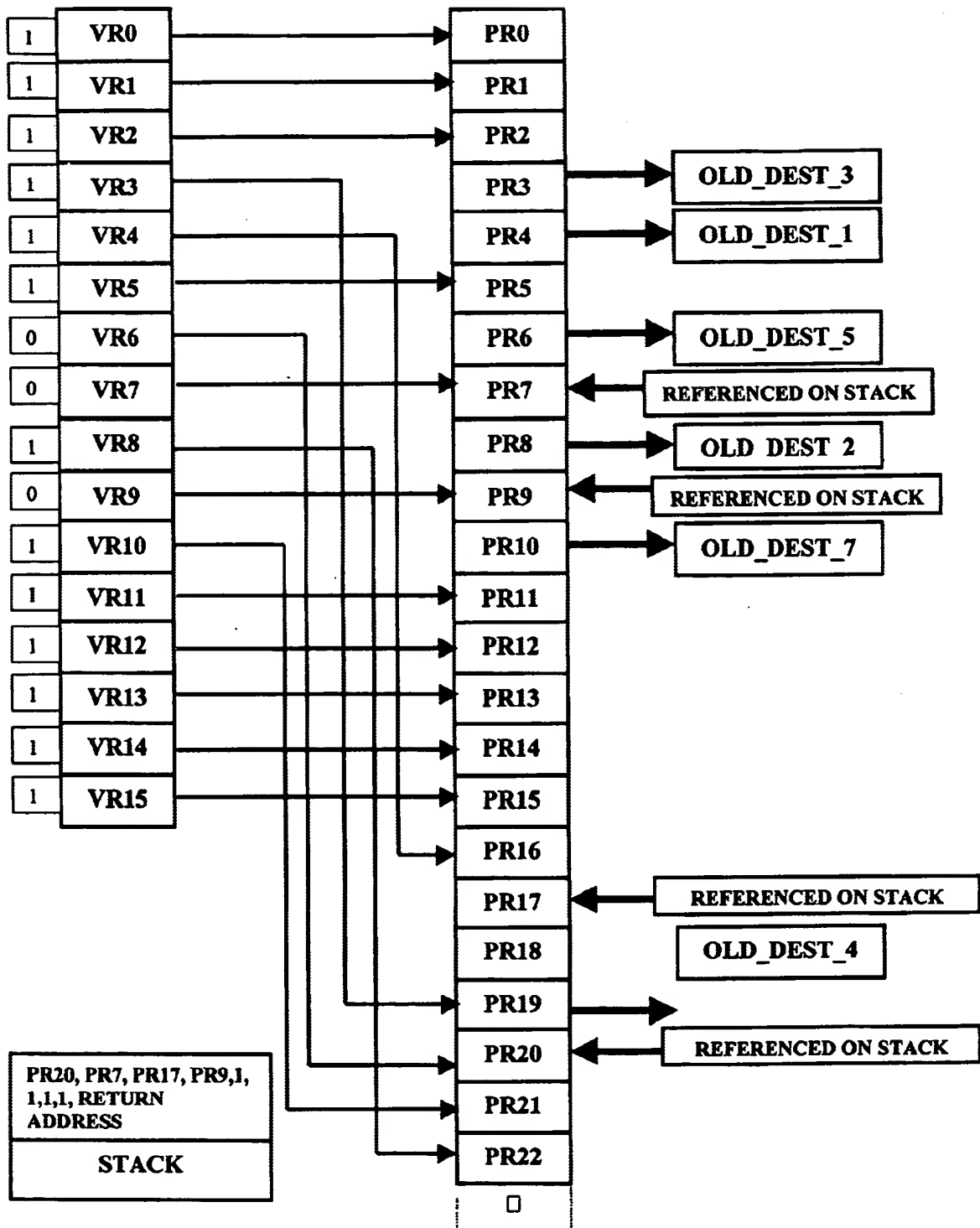
CLOCK 4: DECODE STAGE
INSTRUCTIONS 5 & 6 PHYSICAL REGISTER STATE

FIG. 32



INSTR. 7: ADD VR6, VR3, VR10 maps to ADD PR20, PR19, PR21,
PR10 → OLD_DEST_7

FIG. 33

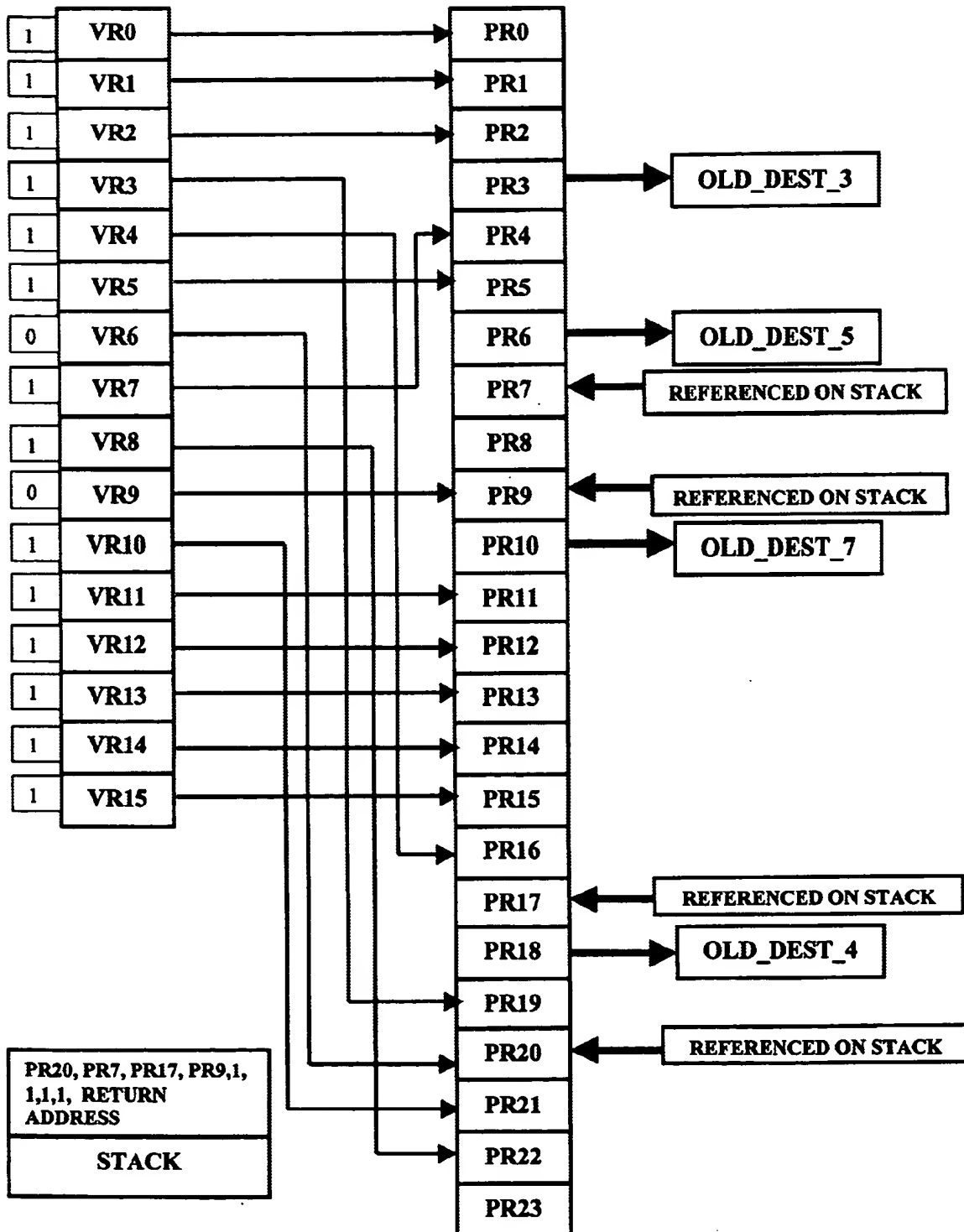


INSTR. 8: SUB VR2, VR3, VR8 maps to SUB PR2, PR19, PR22
1 → DIRTY BIT FOR VR8

FIG. 34

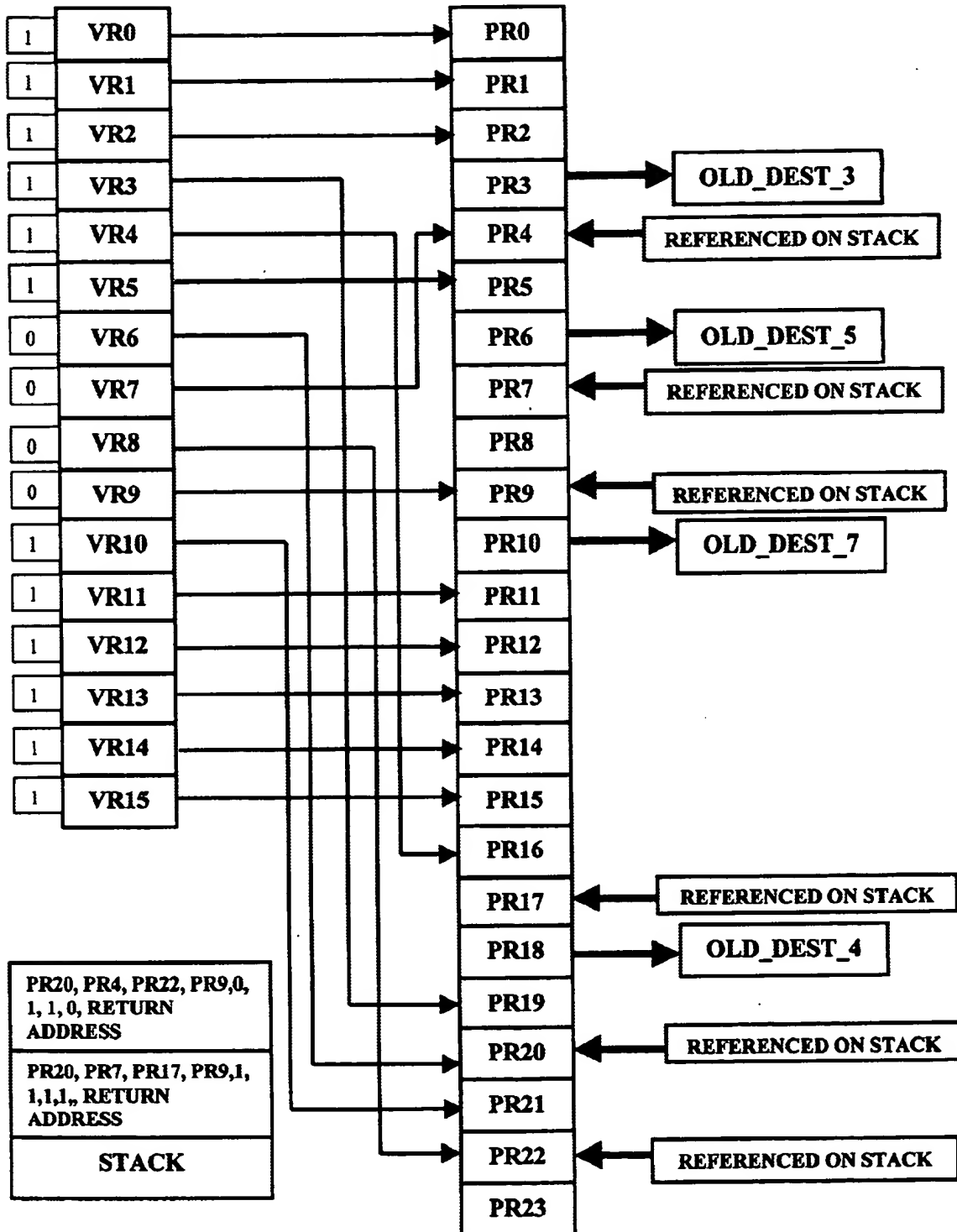
CLOCK 5: DECODE STAGE
INSTRUCTIONS 7 & 8 PHYSICAL REGISTER STATE

FIG. 35



INSTR. 9: `MUL VR8, VR1, VR7` maps to `MUL PR22, PR1, PR4`
 1 → DIRTY BIT FOR VR7

FIG. 36



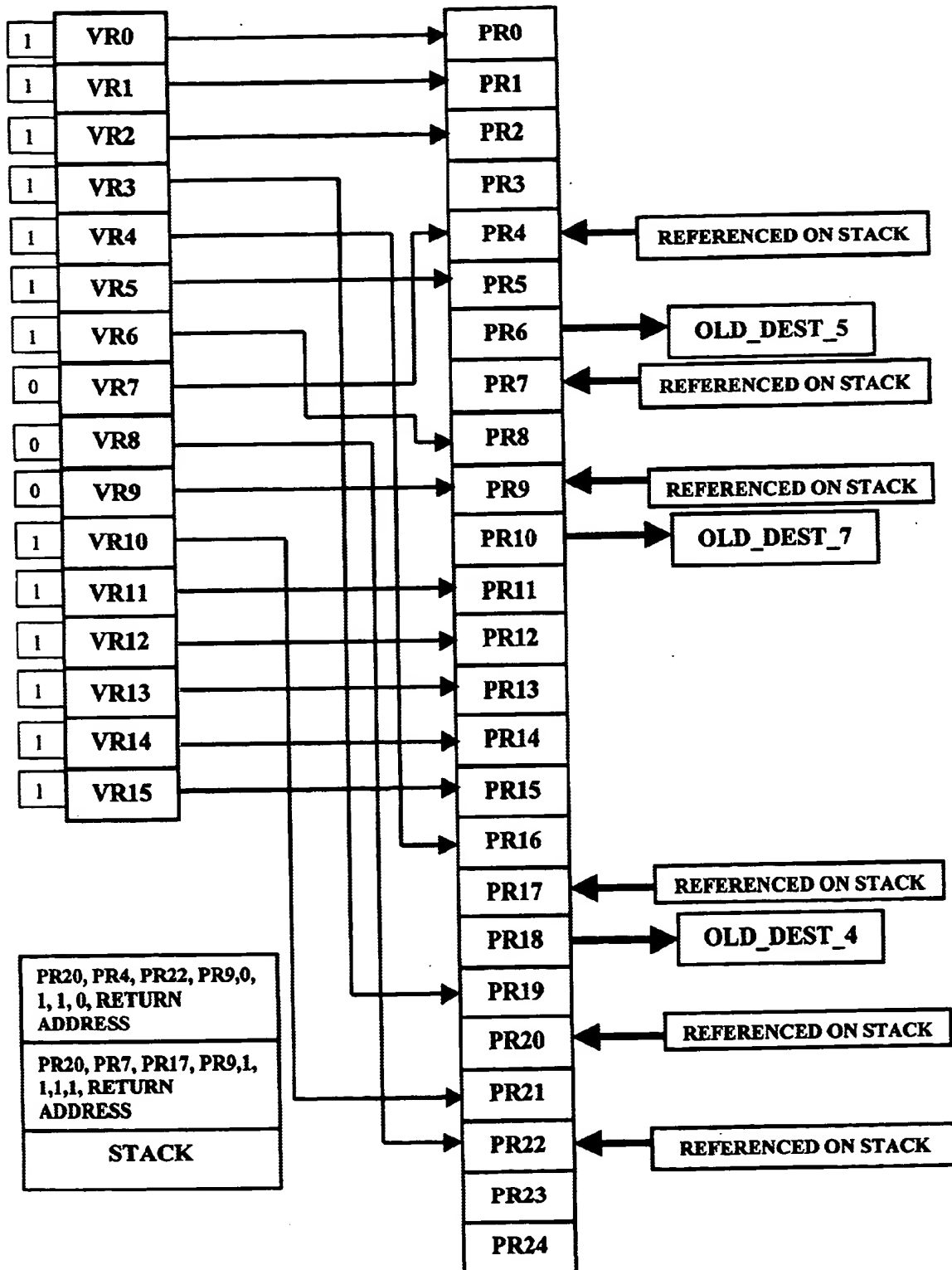
INSTR. 10: CALL B action PUSH PR20, PR4, PR22, PR9, 0, 1, 1, 0,
RETURN ADDRESS, 0000 → DIRTY BITS FOR VR6-9, transfer to B

FIG. 37

CLOCK 6: DECODE STAGE
INSTRUCTIONS 9 & 10 PHYSICAL REGISTER STATE

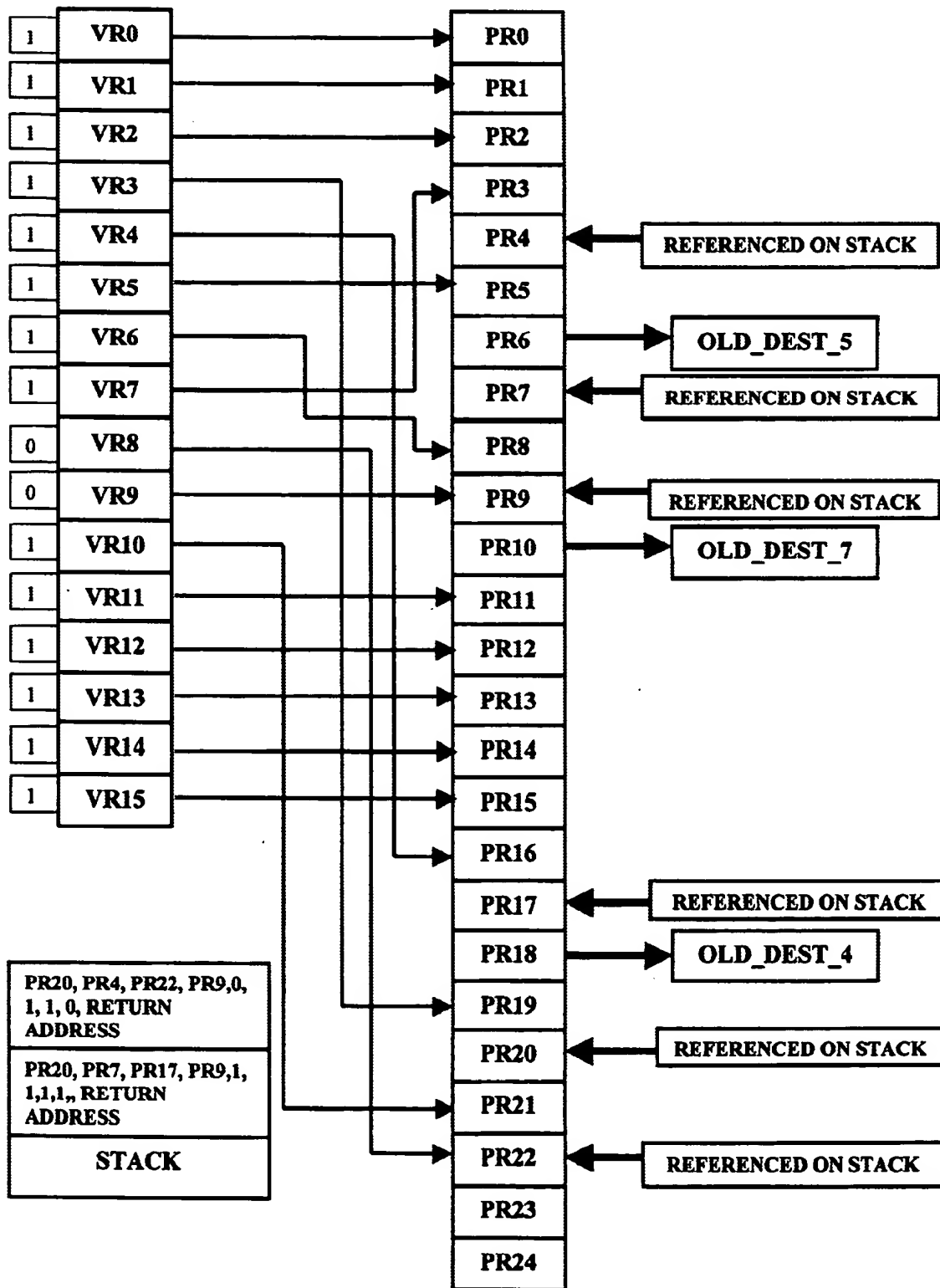
FIG. 38

35/90



INSTR. 11: ADD VR1, VR2, VR6 maps to ADD PR1, PR2, PR8
1 → DIRTY BIT FOR VR6

FIG. 39



INSTR. 12: ADD VR3, VR7, VR7 maps to ADD PR19, PR4, PR3
1 → DIRTY BIT FOR VR7

FIG. 40

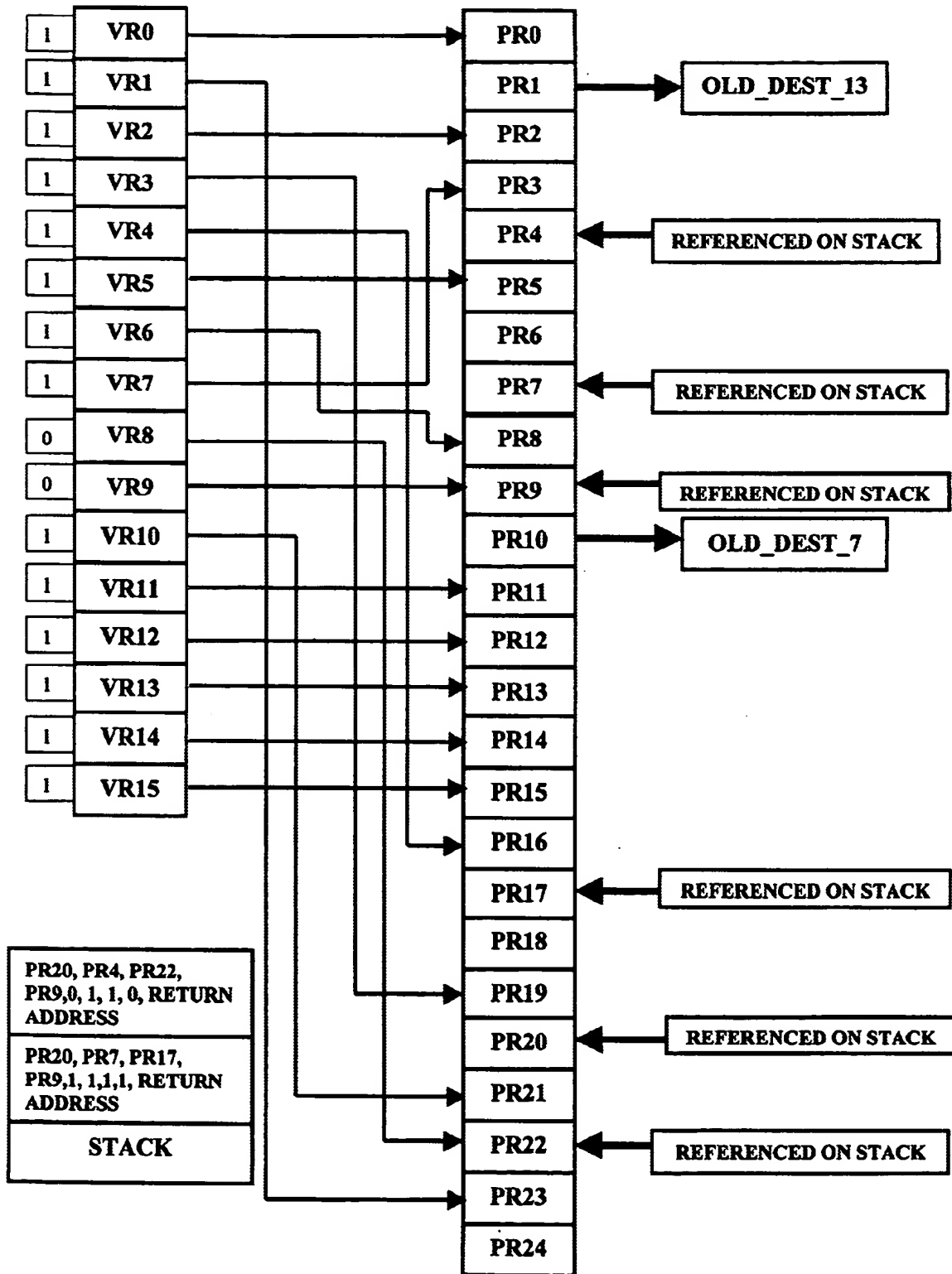
| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|--|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | 1 | EXAMPLE INITIALIZATION |
| 2 | 0 | 1 | 7 | 2 | EXAMPLE INITIALIZATION |
| 3 | 0 | 0 | - | 7 | WAITING FOR INSTRUCTION 12 TO EXECUTE |
| 4 | 0 | 0 | - | - | WAIT FOR INS. 9 TO EXECUTE, REF. SAVED |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | - | WAITING FOR 5 TO RETIRE |
| 7 | 0 | 1 | 17 | - | REFERENCE PREVIOUSLY SAVED ON STACK |
| 8 | 0 | 0 | - | 6 | WAITING FOR INSTRUCTION 11 TO EXECUTE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 0 | 1 | 23 | - | WAITING FOR INSTRUCTION 7 TO RETIRE |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 10 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | - | INSTRUCTION 2 EXECUTED |
| 18 | 0 | 1 | 2 | - | WAITING FOR INSTRUCTION 4 TO RETIRE |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 130 | - | REFERENCE PREVIOUSLY SAVED ON STACK |
| 21 | 0 | 0 | - | 10 | WAITING FOR INSTRUCTION 7 TO EXECUTE |
| 22 | 0 | 0 | - | 8 | WAITING FOR INSTRUCTION 8 TO EXECUTE |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 7: DECODE STAGE
INSTRUCTIONS 11 & 12 PHYSICAL REGISTER STATE

FIG. 41

092253.040301

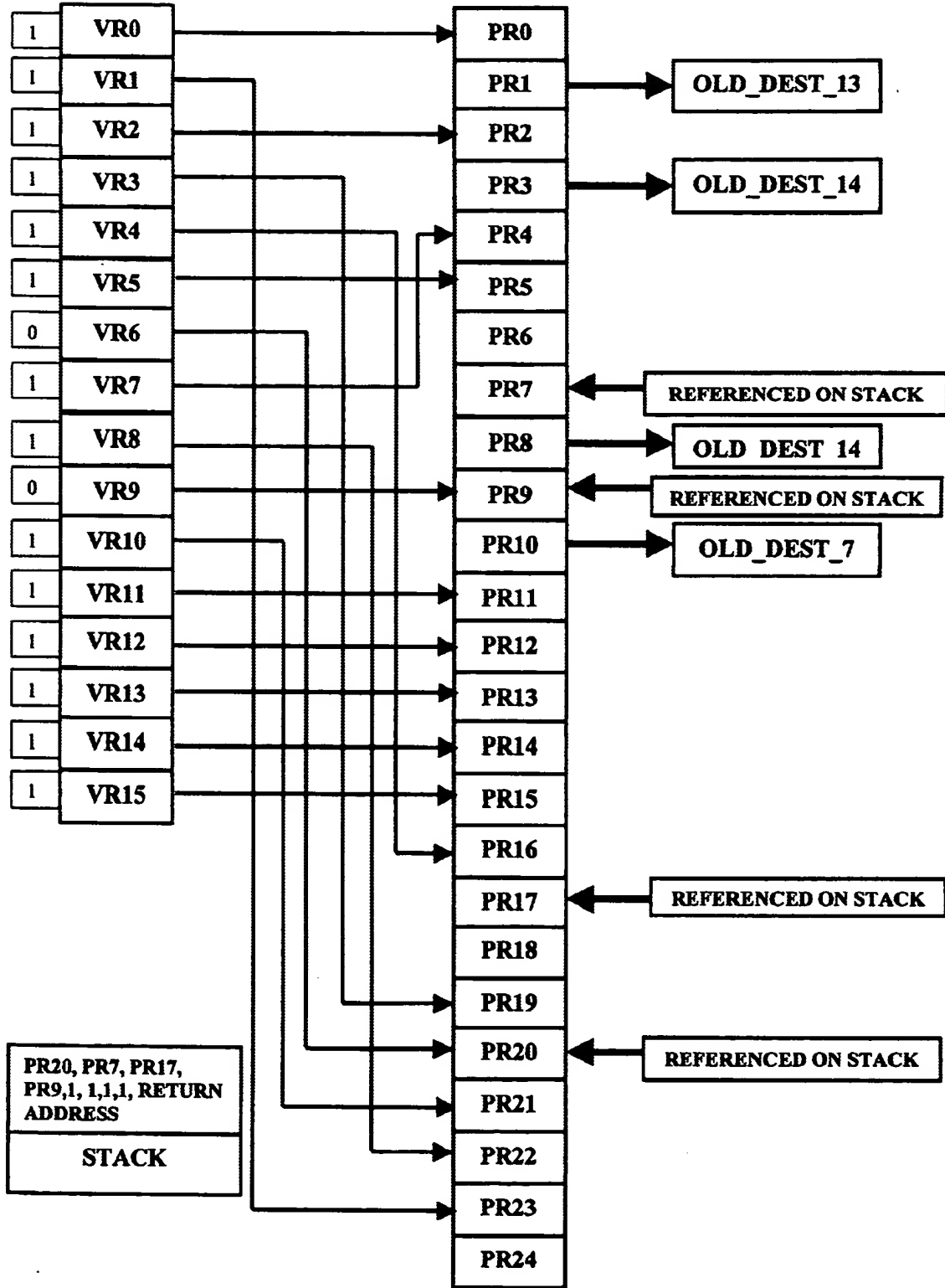
38/90



INSTR. 13: MUL VR6, VR7, VR1 maps to MUL PR8, PR3, PR23
PR1 → OLD_DEST_13

FIG. 42

39/90



INSTR. 14: RET maps to
9'S DIRTY BITS,

POP PR20, PR4, PR22, PR9 → VR6-9, 0110 → VR6-

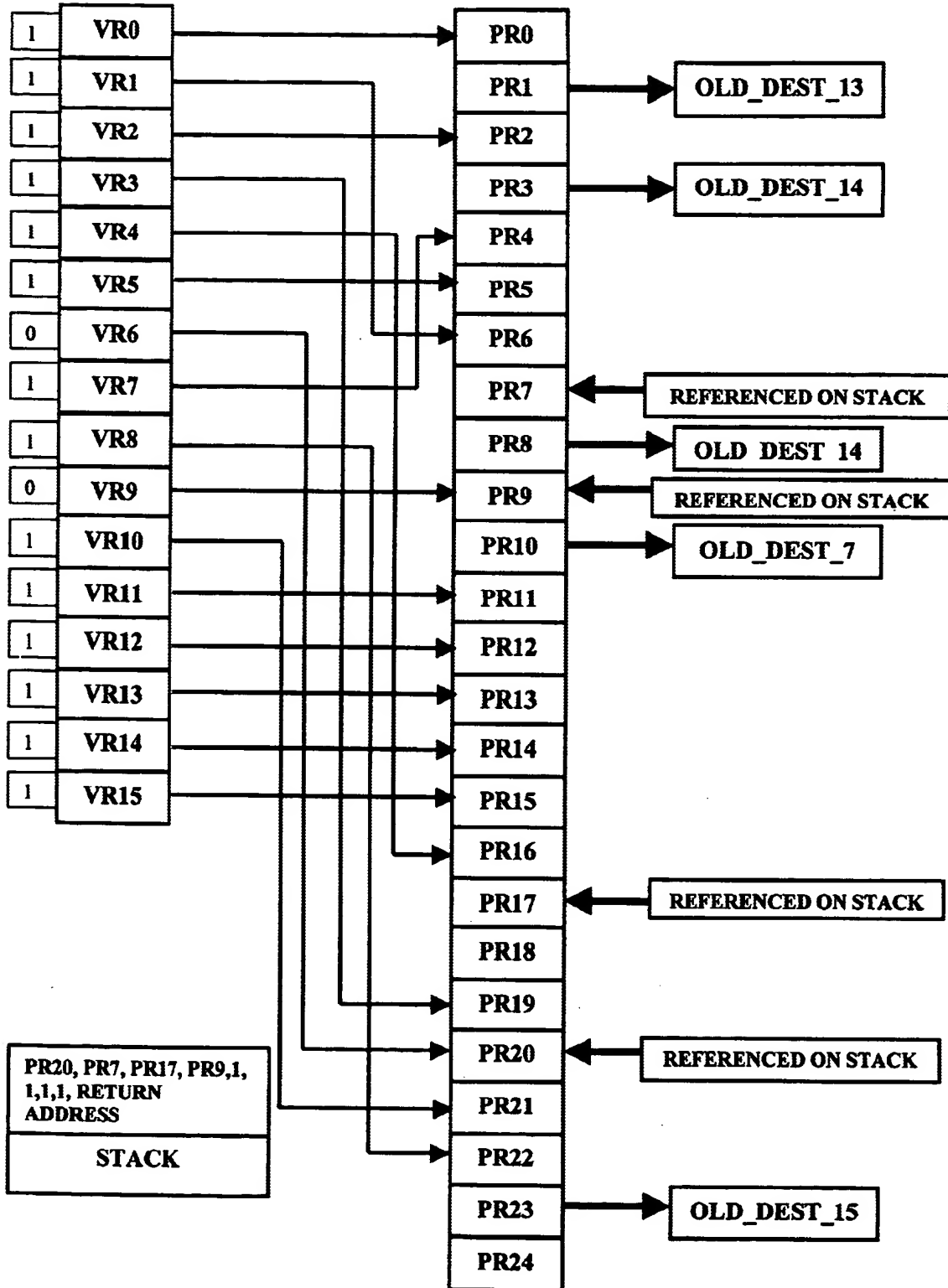
RETURN FROM SUBR. B, PR3 & PR8 → OLD_DEST_14

FIG. 43

CLOCK 8: DECODE STAGE
INSTRUCTIONS 13 & 14 PHYSICAL REGISTER STATE

FIG. 44

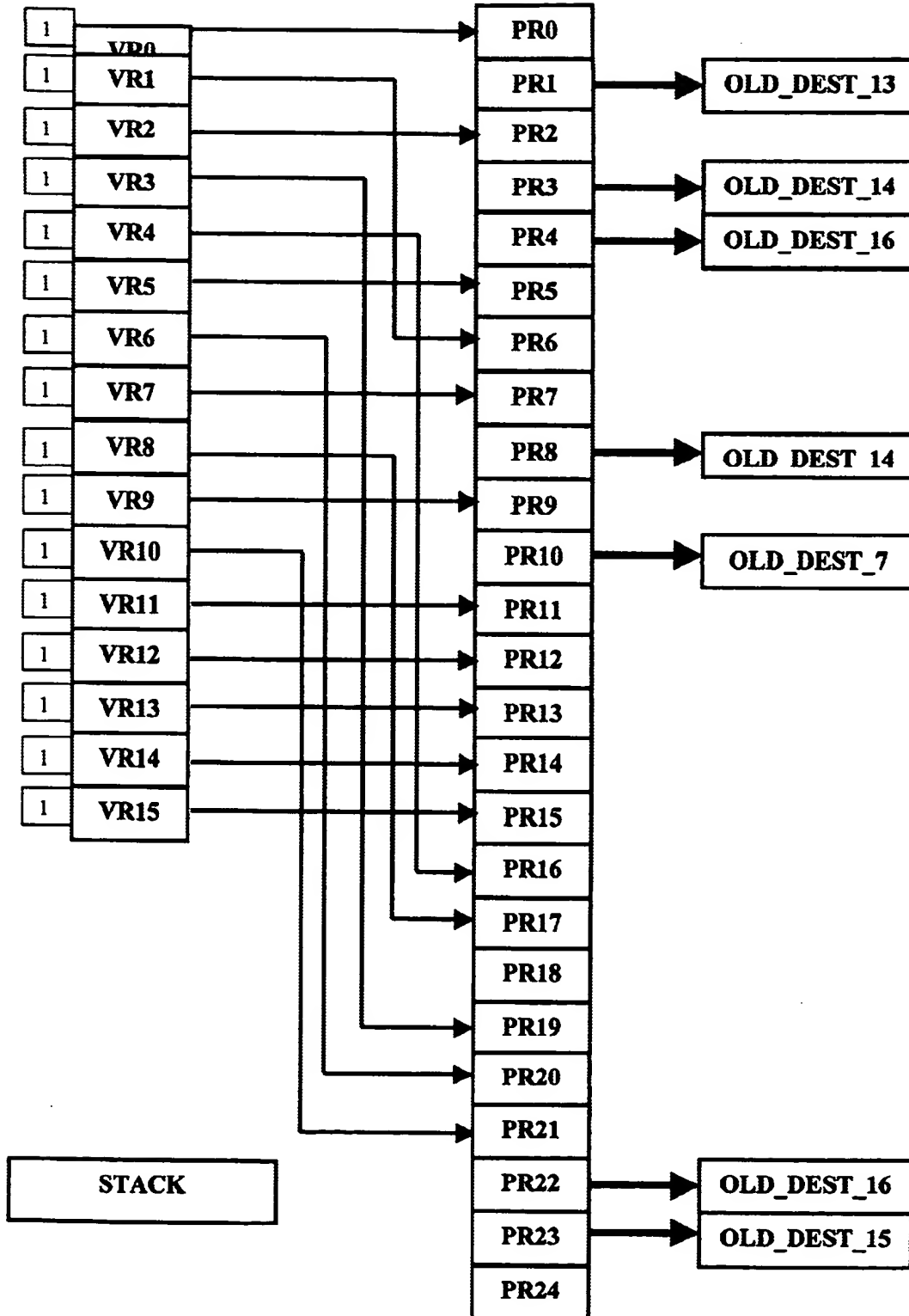
41/90



INSTR. 15: ADD VR8, VR7, VR1 maps to ADD PR22, PR4, PR6
PR23 → OLD_DEST_15

FIG. 45

42/90



INSTR. 16: RET maps to POP PR20, PR7, PR17, PR9 → VR6-9, 1111 → VR6-9'S DIRTY BITS,

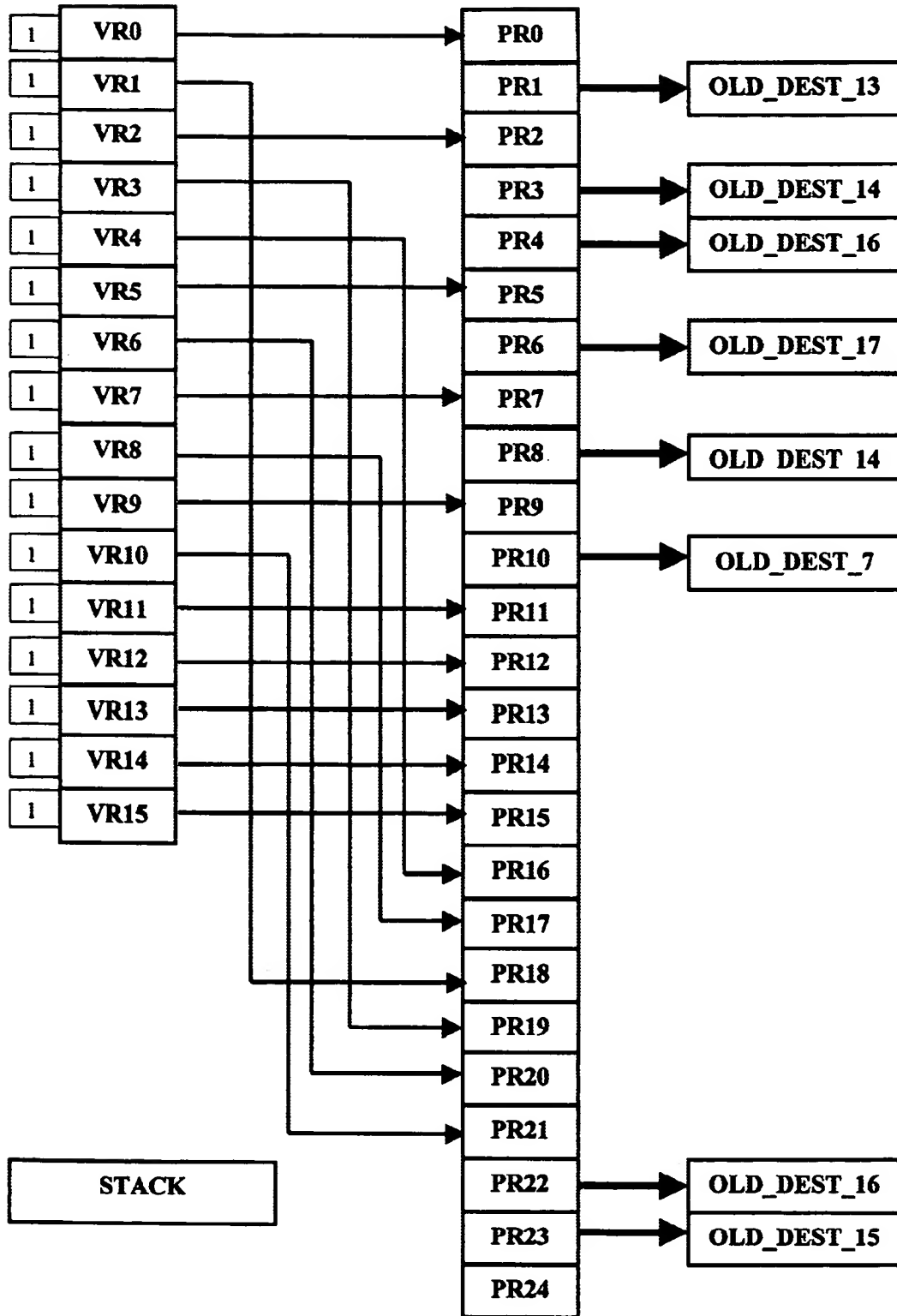
RETURN FROM SUBR. A, PR4 & PR22 → OLD_DEST_16

FIG. 46

CLOCK 9: DECODE STAGE
INSTRUCTIONS 15 & 16 PHYSICAL REGISTER STATE

FIG. 47

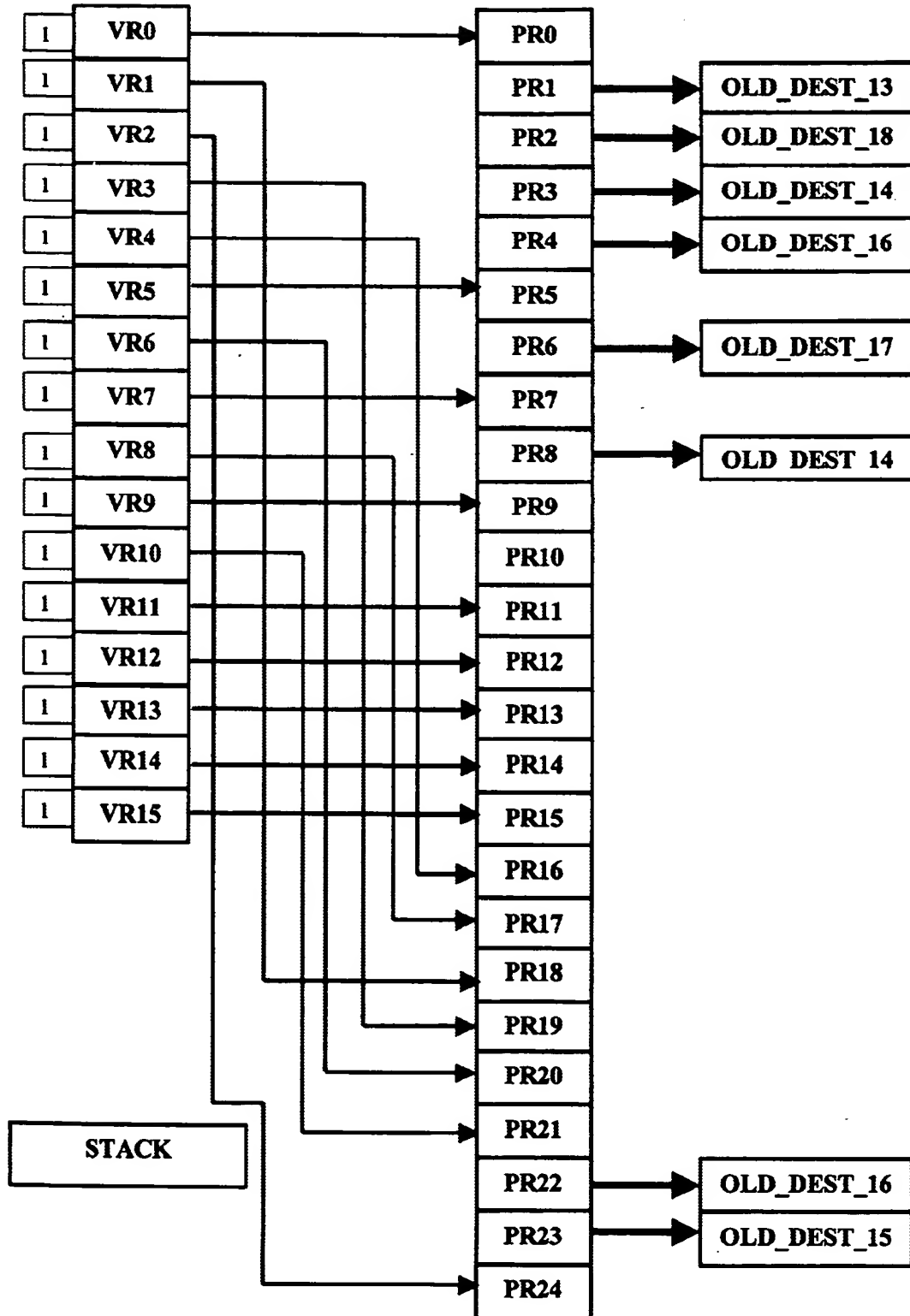
44/90



INSTR. 17: ADD VR8, VR1, VR1 maps to ADD PR17, PR6, PR18
PR6 → OLD_DEST_17

FIG. 48

45/90



INSTR. 18: ADD VR8, VR2, VR2 maps to ADD PR17, PR2, PR24
PR2 → OLD_DEST_18

FIG. 49

CLOCK 10: DECODE STAGE
INSTRUCTIONS 17 & 18 PHYSICAL REGISTER STATE

FIG. 50

CLOCK 11: DECODE STAGE
NO CHANGE IN PHYSICAL REGISTER STATE

FIG. 51

CLOCK 12: DECODE STAGE
PHYSICAL REGISTER STATE

FIG. 52

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|--|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | - | WAITING FOR INSTRUCTION 13 TO RETIRE |
| 2 | 0 | 1 | 7 | - | WAIT FOR INSTRUCTION 18 TO RETIRE |
| 3 | 0 | 1 | -13 | - | WAIT FOR INS. 14 TO RETIRE |
| 4 | 0 | 1 | -25 | - | WAIT FOR INS. 16 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | -30 | - | WAIT FOR INS.17 TO RETIRE |
| 7 | 0 | 1 | 17 | 7 | VR7 REFERENCE RESTORED FROM STACK |
| 8 | 0 | 1 | 12 | - | WAIT FOR INS. 14 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 1 | - | - | - | INSTRUCTION 7 RETIRED |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 10 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | 8 | VR8 REFERENCE RESTORED FROM STACK |
| 18 | 0 | 1 | -134 | 1 | INSTRUCTION 17 EXECUTED |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 130 | 6 | VR6 REFERENCE RESTORED FROM STACK |
| 21 | 0 | 1 | 142 | 10 | INSTRUCTION 7 EXECUTED |
| 22 | 0 | 1 | -5 | - | WAIT FOR INS. 16 TO RETIRE |
| 23 | 0 | 0 | - | - | WAIT FOR INS. 13 TO EXEC. & 15 TO RETIRE |
| 24 | 0 | 1 | -8 | 2 | INSTRUCTION 18 EXECUTED |
| ETC. | 1 | - | - | - | UNALLOCATED |

**CLOCK 13: DECODE STAGE
PHYSICAL REGISTER STATE**

FIG. 53

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|--------------------------------------|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | - | WAITING FOR INSTRUCTION 13 TO RETIRE |
| 2 | 0 | 1 | 7 | - | WAIT FOR INSTRUCTION 18 TO RETIRE |
| 3 | 0 | 1 | -13 | - | WAIT FOR INS. 14 TO RETIRE |
| 4 | 0 | 1 | -25 | - | WAIT FOR INS. 16 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | -30 | - | WAIT FOR INS.17 TO RETIRE |
| 7 | 0 | 1 | 17 | 7 | VR7 REFERENCE RESTORED FROM STACK |
| 8 | 0 | 1 | 12 | - | WAIT FOR INS. 14 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 1 | - | - | - | INSTRUCTION 7 RETIRED |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 10 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | 8 | VR8 REFERENCE RESTORED FROM STACK |
| 18 | 0 | 1 | -134 | 1 | INSTRUCTION 17 EXECUTED |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 130 | 6 | VR6 REFERENCE RESTORED FROM STACK |
| 21 | 0 | 1 | 142 | 10 | INSTRUCTION 7 EXECUTED |
| 22 | 0 | 1 | -5 | - | WAIT FOR INS. 16 TO RETIRE |
| 23 | 0 | 1 | -156 | - | WAIT FOR INS. 15 TO RETIRE |
| 24 | 0 | 1 | -8 | 2 | INSTRUCTION 18 EXECUTED |
| ETC. | 1 | - | - | - | UNALLOCATED |

**CLOCK 14: DECODE STAGE
PHYSICAL REGISTER STATE**

FIG. 54

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|-----------------------------------|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 1 | - | - | - | INSTRUCTION 13 RETIRED |
| 2 | 1 | - | - | - | INSTRUCTION 18 RETIRED |
| 3 | 1 | - | - | - | INSTRUCTION 14 RETIRED |
| 4 | 1 | - | - | - | INSTRUCTION 16 RETIRED |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 1 | - | - | - | INSTRUCTION 17 RETIRED |
| 7 | 0 | 1 | 17 | 7 | VR7 REFERENCE RESTORED FROM STACK |
| 8 | 1 | - | - | - | INSTRUCTION 14 RETIRED |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 1 | - | - | - | INSTRUCTION 7 RETIRED |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 10 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | 8 | VR8 REFERENCE RESTORED FROM STACK |
| 18 | 0 | 1 | -134 | 1 | INSTRUCTION 17 EXECUTED |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 130 | 6 | VR6 REFERENCE RESTORED FROM STACK |
| 21 | 0 | 1 | 142 | 10 | INSTRUCTION 7 EXECUTED |
| 22 | 1 | - | - | - | INSTRUCTION 16 RETIRED |
| 23 | 1 | 0 | - | - | INSTRUCTION 15 RETIRED |
| 24 | 0 | 1 | -8 | 2 | INSTRUCTION 18 EXECUTED |
| ETC. | 1 | - | - | - | UNALLOCATED |

**CLOCK 15: DECODE STAGE
PHYSICAL REGISTER STATE**

FIG. 55

```

A:  ADD VR6, VR3, VR10  Subroutine uses Arguments VR1 and VR2
    SUB VR2, VR3, VR8
    MUL VR8, VR1, VR7
    CALL B, 2, 8        ; Bind Arg2 to new Arg1 and bind VR8 to new Arg2
    ADD VR8, VR7, VR1
    RET                 ; Restore previous argument bindings

B:  ADD VR1, VR2, VR6    ; Subroutine uses Arguments VR1 and VR2
    ADD VR3, VR7, VR7
    MUL VR6, VR7, VR1
    RET                 ; Restore previous argument bindings

```

start of example execution

```

C:  □
    □
    □
    ADD VR0, VR0, VR4
    LIM VR8, #22
    SUB VR3, VR0, VR3
    ADD VR4, VR3, VR3
    MUL VR4, VR5, VR6
    CALL A, 6, 8        ; Bind VR6 to new Arg1 and bind VR8 to new Arg2
    ADD VR8, VR0, VR0
    ADD VR8, VR6, VR6
    □
    □
    □

```

end of example execution

EXAMPLE PROGRAM

FIG. 56

050253.040301

CLOCK 1: DECODE STAGE
INITIAL PHYSICAL REGISTER STATE

FIG. 57

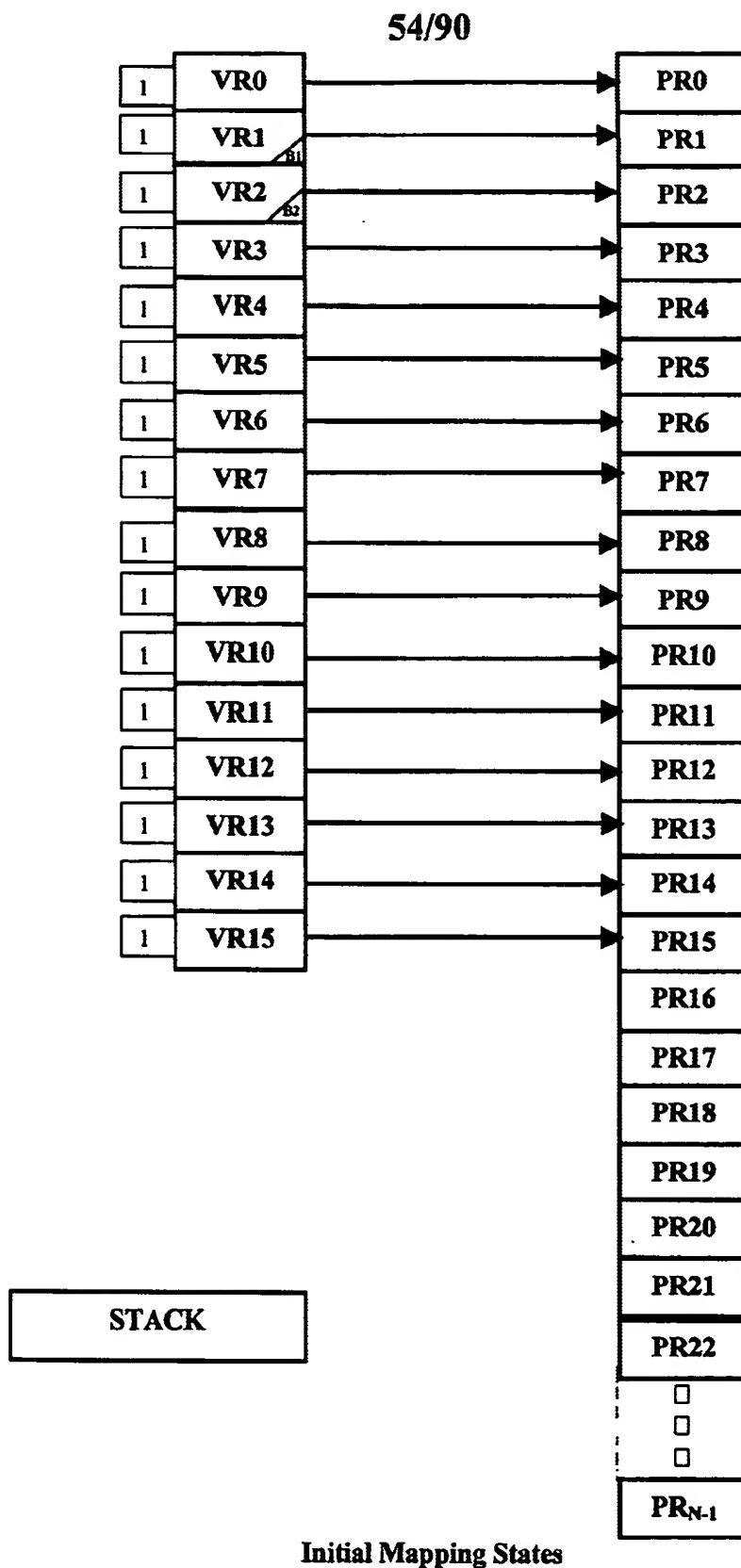


FIG. 58

| INSTRUC- TION # | INSTRUCTION | DESCRIPTION | EFFECT OF INSTRUCTION |
|--------------------|--------------------|-------------------------------|--|
| 1 | ADD VR0, VR0, VR4 | VR0 + VR0 → VR4 | (3 + 3) → 6 → VR4 |
| 2 | LIM VR8, #22 | 22 → VR8 | 22 ₁₀ → VR8 |
| 3 | SUB VR3, VR0, VR3 | VR3 - VR0 → VR3 | (9 - 3) → 6 → VR3 |
| 4 | ADD VR4, VR3, VR3 | VR4 + VR3 → VR3 | (6 + 6) → 12 → VR3 |
| 5 | MUL VR4, VR5, VR6 | VR4 * VR5 → VR6 | (6 * 13) → 78 → VR6 |
| 6 | CALL A, VR6, VR8 | CALL subroutine A(Arg1, Arg2) | Arg1 □ VR6, Arg2 □ VR8, VR6—VR9 scratch registers; VR1 □ Arg1, VR2 □ Arg2. |
| 7 | ADD VR1, VR3, VR10 | VR1 + VR3 → VR10 | (78 + 12) → 90 → VR10 (Uses C program's VR6 as source) |
| 8 | SUB VR2, VR3, VR8 | VR2 - VR3 → VR8 | (22 - 12) → 10 → VR8 (Uses C programs' VR8 as source) |
| 9 | MUL VR8, VR1, VR7 | VR8 * VR1 → VR7 | (10 * 78) → 780 → VR7 (use VR7 as scratch register) |
| 10 | CALL B, VR2, VR8 | CALL subroutine B(Arg1, Arg2) | Arg1 □ A's Arg2, Arg2 □ VR8, VR6—VR9 scratch registers; VR1 □ Arg1, VR2 □ Arg2. |
| 11 | ADD VR1, VR2, VR6 | VR1 + VR2 → VR6 | (22 + 10) → 32 → VR6 (Uses C program's VR8 as source, A program's VR8 as source, and uses VR6 as scratch register) |
| 12 | ADD VR3, VR7, VR7 | VR3 + VR7 → VR7 | (12 + 780) → 792 → VR7 (use VR7 as scratch register) |
| 13 | MUL VR6, VR7, VR1 | VR6 * VR7 → VR1 | (32 * 792) → 25344 → VR1 (Uses C program's VR8 as destination) |
| 14 | RET | RETURN | restore value of 78 to VR6, 780 to VR7, VR1 link to C's VR6, and VR2 link to C's VR8. |
| 15 | ADD VR8, VR7, VR1 | VR8 + VR7 → VR1 | (10 + 780) → 790 → VR1 (Uses C program's VR6 as destination) |
| 16 | RET | RETURN | restore value of 790 to VR6, 17 to VR7, 25344 to VR8, and VR1 and VR2 links to VRs in Program that Called C. |
| 17 | ADD VR8, VR0, VR0 | VR8 + VR0 → VR0 | (25344 + 3) → 25347 → VR0 |
| 18 | ADD VR8, VR6, VR6 | VR8 + VR6 → VR6 | (25344 + 790) → 26134 → VR6 |

EXAMPLE INSTRUCTION FLOW

FIG. 59

| INSTRUCTION NUMBER | VIRTUAL REGISTER NUMBER: | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----------------------|--------------------------|----------------|-------|----|----|----|----|----|-------|-----|-------|----|----|----|----|----|----|----|
| | INSTRUCTION | INITIAL VALUE: | | | | | | | | | | | | | | | | |
| 1 | ADD VR0, VR0, VR4 | 3 | 5 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 2 | LIM VR8, #22 | 3 | 5 | 5 | 7 | 9 | 6 | 13 | 15 | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 3 | SUB VR3, VR0, VR3 | 3 | 5 | 5 | 7 | 6 | 6 | 13 | 15 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 4 | ADD VR4, VR3, VR3 | 3 | 5 | 5 | 7 | 12 | 6 | 13 | 15 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 5 | MUL VR4, VR5, VR6 | 3 | 5 | 5 | 7 | 12 | 6 | 13 | 78 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 6 | CALL A, VR6, VR8 | 3 | 78 | 22 | 12 | 6 | 6 | 13 | 78 | 17 | 22 | 21 | 23 | 25 | 27 | 29 | 31 | 33 |
| 7 | ADD VR1, VR3, VR10 | 3 | 78 | 22 | 12 | 6 | 6 | 13 | 78 | 17 | 22 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 8 | SUB VR2, VR3, VR8 | 3 | 78 | 22 | 12 | 6 | 6 | 13 | 78 | 17 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 9 | MUL VR8, VR1, VR7 | 3 | 78 | 22 | 12 | 6 | 6 | 13 | 78 | 780 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 10 | CALL B, VR2, VR8 | 3 | 22 | 10 | 12 | 6 | 6 | 13 | 78 | 780 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 11 | ADD VR1, VR2, VR6 | 3 | 22 | 10 | 12 | 6 | 6 | 13 | 32 | 780 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 12 | ADD VR3, VR7, VR7 | 3 | 22 | 10 | 12 | 6 | 6 | 13 | 32 | 792 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 13 | MUL VR6, VR7, VR1 | 3 | 25344 | 10 | 12 | 6 | 6 | 13 | 32 | 792 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 14 | RET | 3 | 78 | 22 | 12 | 6 | 6 | 13 | 78 | 780 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 15 | ADD VR8, VR7, VR1 | 3 | 790 | 22 | 12 | 6 | 6 | 13 | 78 | 780 | 10 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 16 | RET | 3 | 5 | 7 | 12 | 6 | 6 | 13 | 790 | 17 | 25344 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 17 | ADD VR8, VR0, VR0 | 25347 | 5 | 7 | 12 | 6 | 6 | 13 | 790 | 17 | 25344 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |
| 18 | ADD VR8, VR6, VR6 | 25347 | 5 | 7 | 12 | 6 | 6 | 13 | 25134 | 17 | 25344 | 21 | 90 | 25 | 27 | 29 | 31 | 33 |

CONTENTS OF VIRTUAL REGISTERS AS INSTRUCTIONS EXECUTE

FIG. 60

57/90

| | | | |
|----------------|----------------------|-----------------------|--|
| <u>Clock 1</u> | Fetch instr. 1, 2. | | |
| <u>Clock 2</u> | Fetch instr. 3, 4; | Decode instr. 1, 2. | |
| <u>Clock 3</u> | Fetch instr. 5, 6; | Decode instr. 3, 4; | Read regs. PR0 for instr. 1. |
| <u>Clock 4</u> | Fetch instr. 7, 8; | Decode instr. 5, 6; | Read regs. PR0, PR3 for instr. 3; |
| | | | Execute instr. 1, 2 and store results in PR16, PR17 respectively. Execute instr. 6 (CALL A) including binding VR1 to VR6 and VR2 to VR8. |
| <u>Clock 5</u> | Fetch instr. 9, 10; | Decode instr. 7, 8; | Read regs. PR5, PR16 for instr. 5; |
| <u>Clock 6</u> | Fetch instr. 11, 12; | Decode instr. 9, 10; | Read regs. PR16, PR18 for instr. 4; |
| | | | Execute instr. 5 and store result in PR20; Execute instr. 10 (CALL B) including binding VR1 to VR2 and VR2 to VR8. Retire instr. 3. |
| <u>Clock 7</u> | Fetch instr. 13, 14; | Decode instr. 11, 12; | Execute instr. 4 and store result in PR19. |
| <u>Clock 8</u> | Fetch instr. 15, 16; | Decode instr. 13, 14; | Execute instr. 14(Return) including restoring bindings to that for "A". Retire instr. 4, 5, 6. |

Clock by Clock Pipeline Description

FIG. 61A

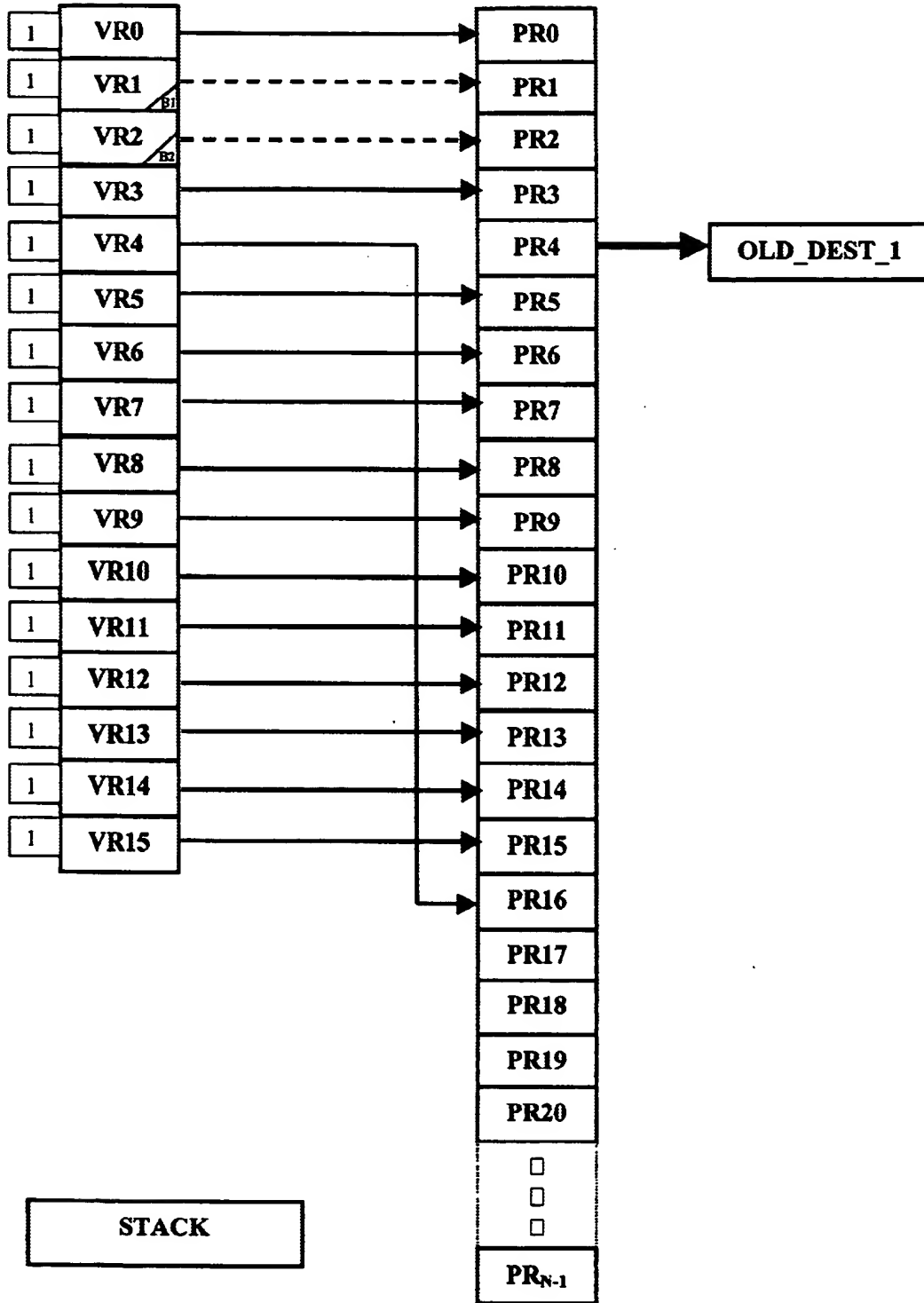
58/90

| | |
|-----------------|---|
| <u>Clock 9</u> | Fetch instr. 17, 18; Decode instr. 15, 16; Read regs. PR17, PR20, PR22 for instr. 9 and 11; Execute instr. 7 and 8 and store results in PR21 and PR22 respectively. Execute instr. 16(Return) including restoring bindings to that for the "C". |
| <u>Clock 10</u> | Decode instr. 17, 18; Execute instr. 9 and 11 and store results in PR4 and PR8 respectively. |
| <u>Clock 11</u> | Read regs. PR4, PR19, PR22 for instr. 12 and 15; Retire instr. 9, 10, 11. |
| <u>Clock 12</u> | Read regs. PR0, PR6, PR23 for instr. 17 and 18; Execute instr. 12, 15 and store results in PR3 and PR6 respectively. |
| <u>Clock 13</u> | Read regs. PR3, PR8 for instr. 13; Execute instr. 17, 18 and store results in PR18 and PR24 respectively; Retire instr. 12. |
| <u>Clock 14</u> | Execute instr. 13 and store results in PR23. |
| <u>Clock 15</u> | Retire instr. 13, 14, 15, 16, 17, 18. |

Clock by Clock Pipeline Description

FIG. 61B

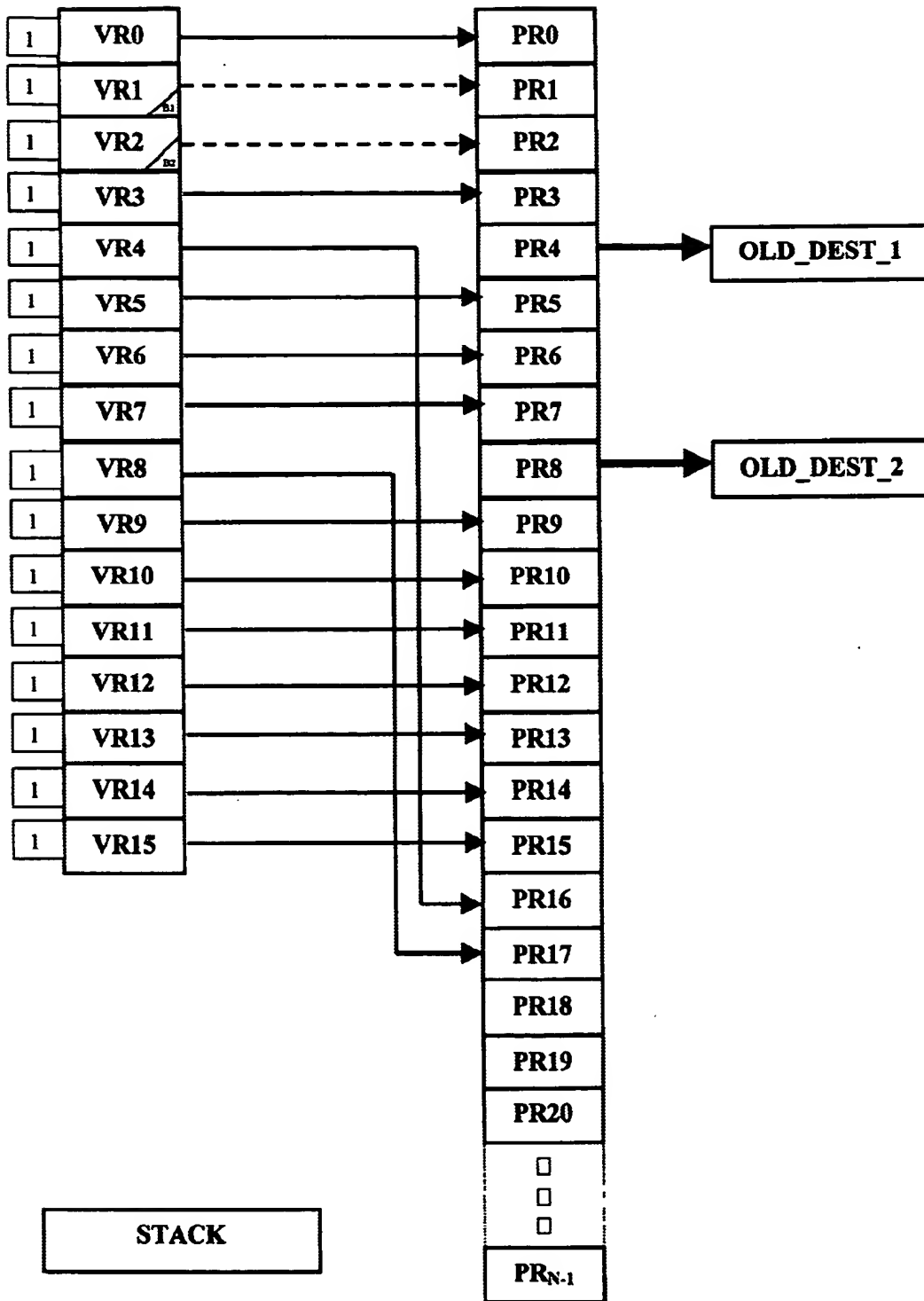
59/90



INSTR. 1: ADD VR0, VR0, VR4 maps to PR0 + PR0 → PR16,
PR4 → OLD_DEST_1

FIG. 62

60/90



INSTR. 2: LIM VR8, #22 maps to LIM PR17, #22,
PR8 → OLD_DEST_2

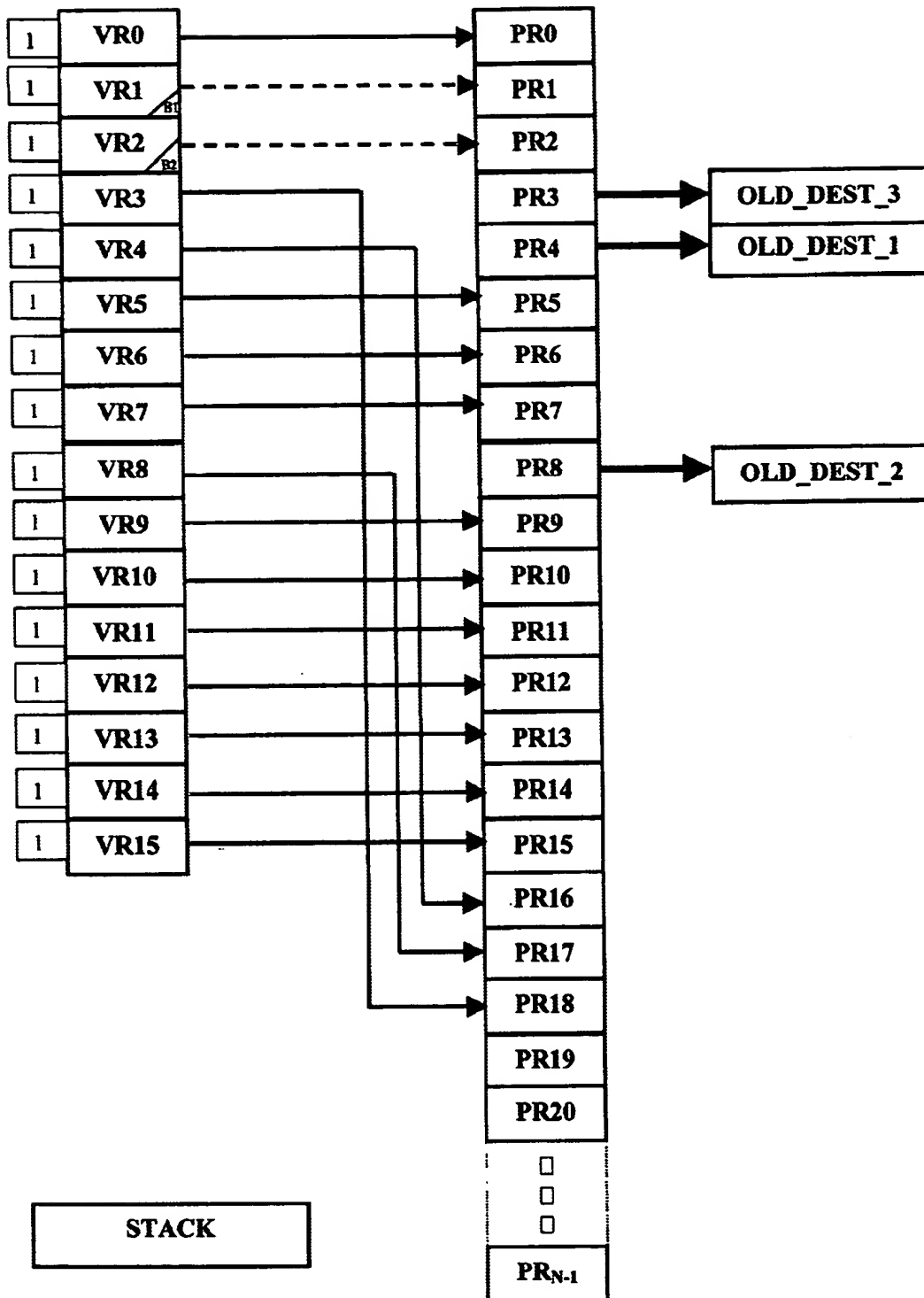
FIG. 63

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|--------------------------------------|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | 1 | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | 2 | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 0 | 1 | 9 | 3 | EXAMPLE INITIALIZATION |
| 4 | 0 | 1 | 11 | - | WAITING FOR INSTRUCTION 1 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | 6 | EXAMPLE INITIALIZATION |
| 7 | 0 | 1 | 17 | 7 | EXAMPLE INITIALIZATION |
| 8 | 0 | 1 | 19 | - | WAITING FOR INSTRUCTION 2 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 0 | 1 | 23 | 10 | EXAMPLE INITIALIZATION |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 0 | - | 4 | WAITING FOR INSTRUCTION 1 TO EXECUTE |
| 17 | 0 | 0 | - | 8 | WAITING FOR INSTRUCTION 2 TO EXECUTE |
| 18 | 1 | - | - | - | UNALLOCATED |
| 19 | 1 | - | - | - | UNALLOCATED |
| 20 | 1 | - | - | - | UNALLOCATED |
| 21 | 1 | - | - | - | UNALLOCATED |
| 22 | 1 | - | - | - | UNALLOCATED |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 2: DECODE STAGE
INSTRUCTIONS 1 & 2 PHYSICAL REGISTER STATE

FIG. 64

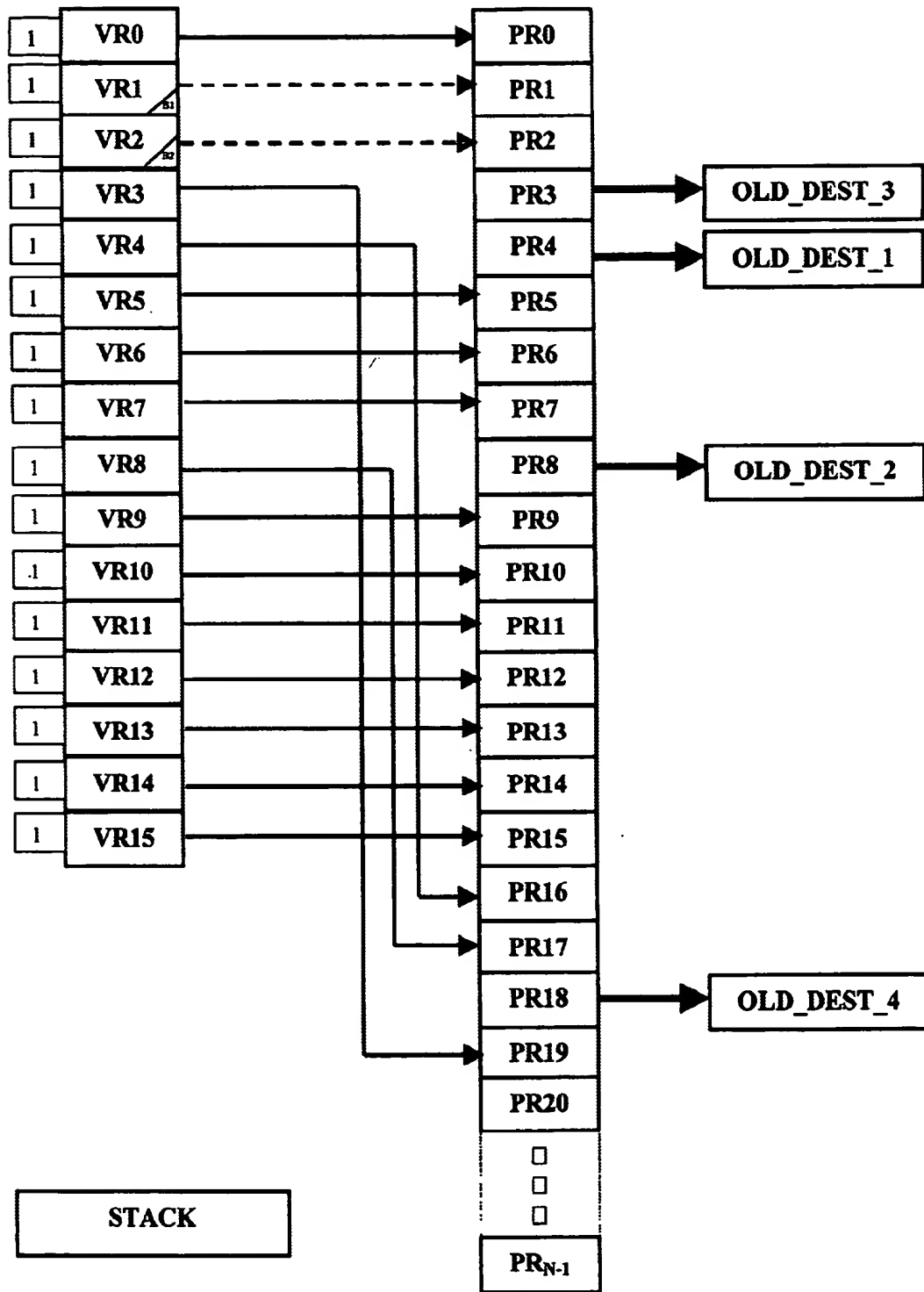
62/90



INSTR. 3: SUB VR3, VR0, VR3 maps to SUB PR3, PR0, PR18,
PR3 → OLD_DEST_3

FIG. 65

63/90



INSTR. 4:

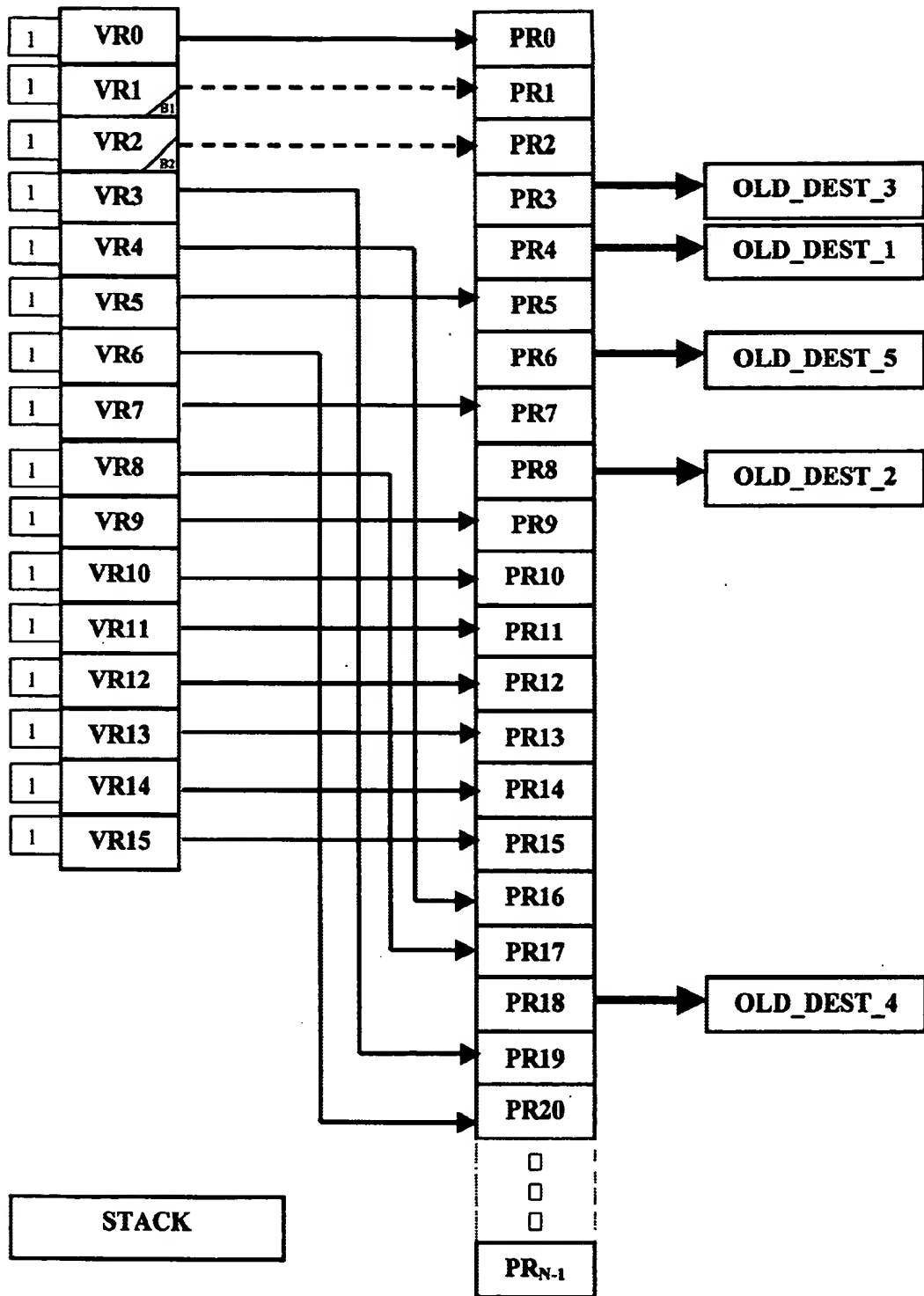
ADD VR4, VR3, VR3
PR19,

maps to ADD PR16, PR18,
PR18 → OLD_DEST_4

FIG. 66

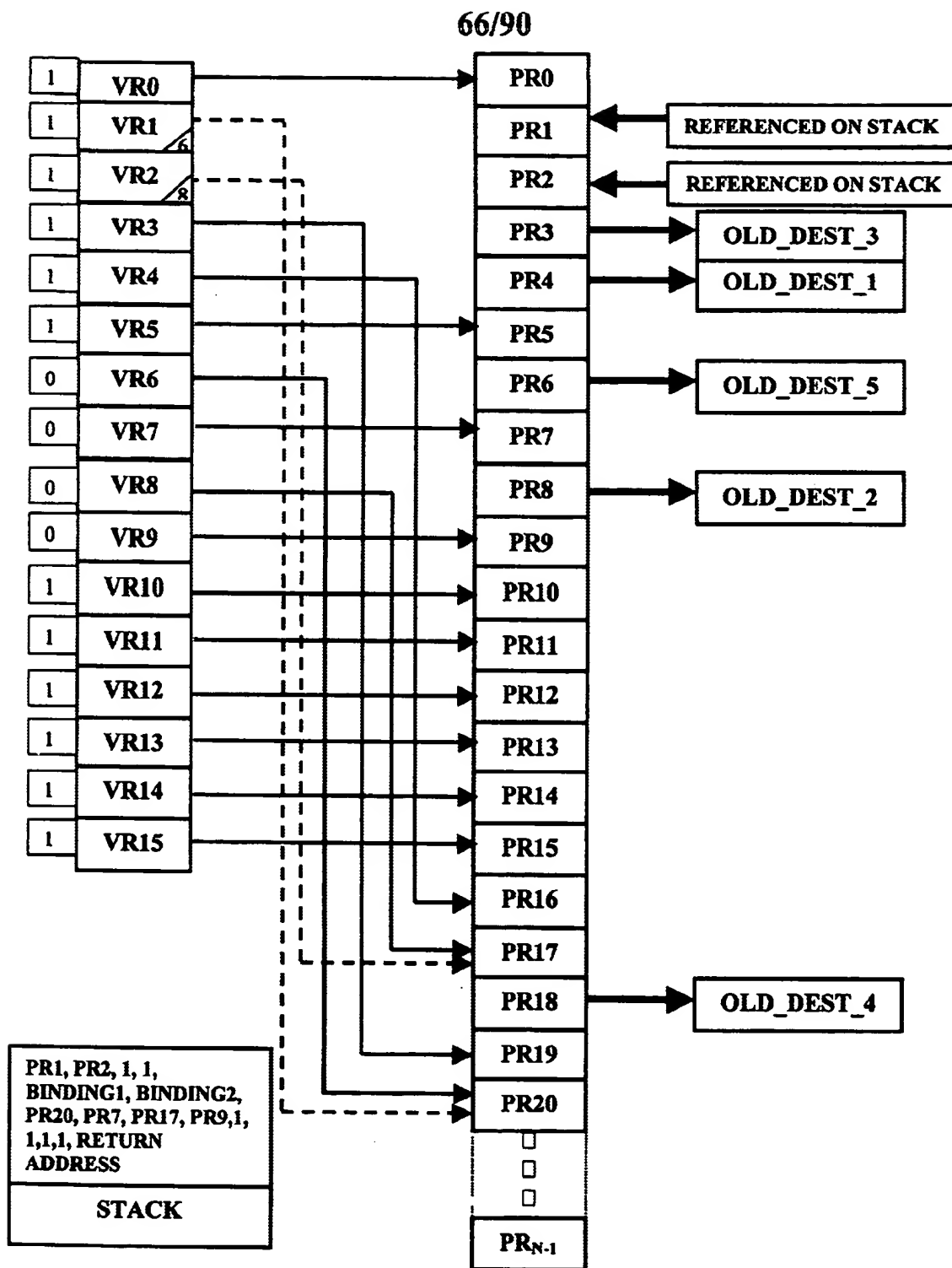
CLOCK 3: DECODE STAGE
INSTRUCTIONS 3 & 4 PHYSICAL REGISTER STATE

FIG. 67



INSTR. 5: MUL VR4, VR5, VR6 maps to MUL PR16, PR5, PR20,
PR6 → OLD_DEST_5

FIG. 68



INSTR. 6: CALL A, VR6, VR8 action PUSH PR1, PR2, 1, 1, BINDING1, BINDING2, PR20, PR7, PR17, PR9, 1, 1, 1, 1, RETURN ADDRESS; BINDVR6_PR20, BINDVR8_PR17, DIRTY BITS FOR VR6&8 → DIRTY BITS FOR VR1&2, 0000 → DIRTY BITS FOR VR6-9, transfer to A

FIG. 69

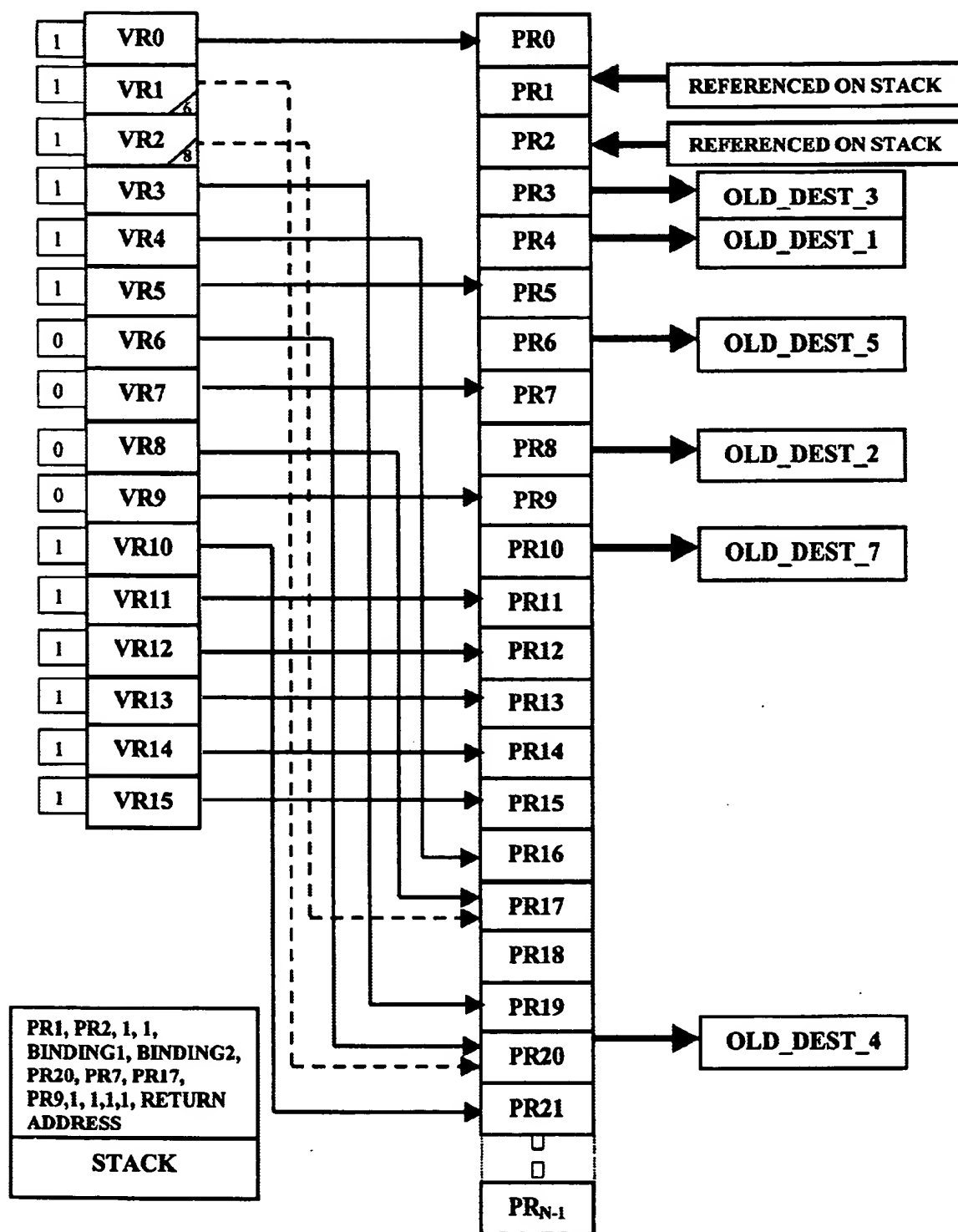
67/90

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|------|---|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | - | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | - | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 0 | 1 | 9 | - | WAITING FOR INSTRUCTION 3 TO RETIRE |
| 4 | 0 | 1 | 11 | - | WAITING FOR INSTRUCTION 1 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | - | WAITING FOR 5 TO RETIRE |
| 7 | 0 | 1 | 17 | 7 | REFERENCED ON STACK |
| 8 | 0 | 1 | 19 | - | WAITING FOR INSTRUCTION 2 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | REFERENCED ON STACK |
| 10 | 0 | 1 | 23 | 10 | EXAMPLE INITIALIZATION |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | 8, 2 | INS. 2 EXECUTED, REFERENCED. ON STACK |
| 18 | 0 | 0 | - | - | WAITING FOR INST. 3 TO EXECUTE & RETIRE |
| 19 | 0 | 0 | - | 3 | WAITING FOR INSTRUCTION 4 TO EXECUTE |
| 20 | 0 | 0 | - | 6, 1 | WAIT FOR INS. 5 TO EXEC., REF'D. ON STACK |
| 21 | 1 | - | - | - | UNALLOCATED |
| 22 | 1 | - | - | - | UNALLOCATED |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

**CLOCK 4: DECODE STAGE
INSTRUCTIONS 5 & 6 PHYSICAL REGISTER STATE**

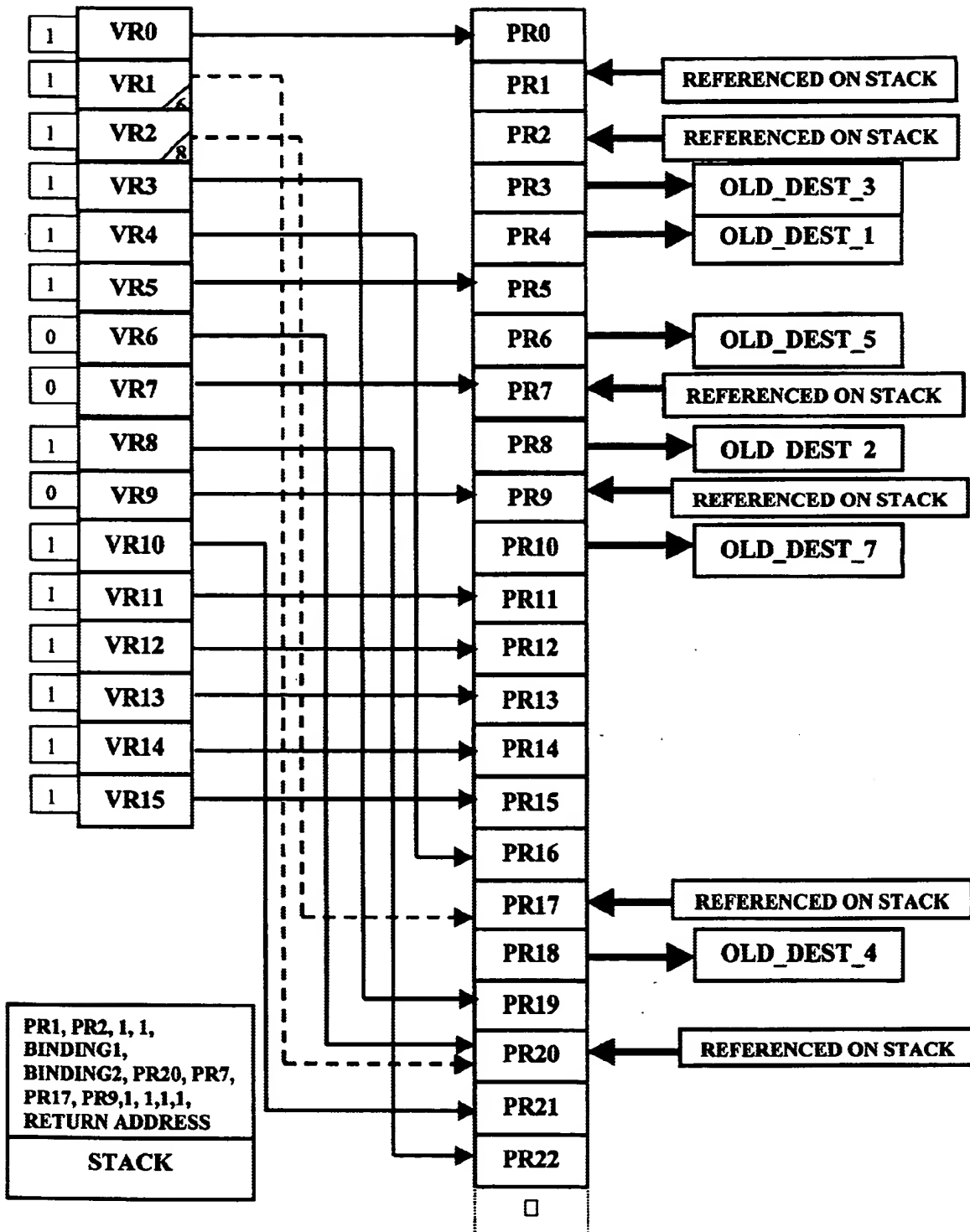
FIG. 70

00825753.040301
10E040" 25752860



INSTR. 7: ADD VR1, VR3, VR10 maps to ADD PR20, PR19, PR21,
PR10 → OLD_DEST_7

FIG. 71



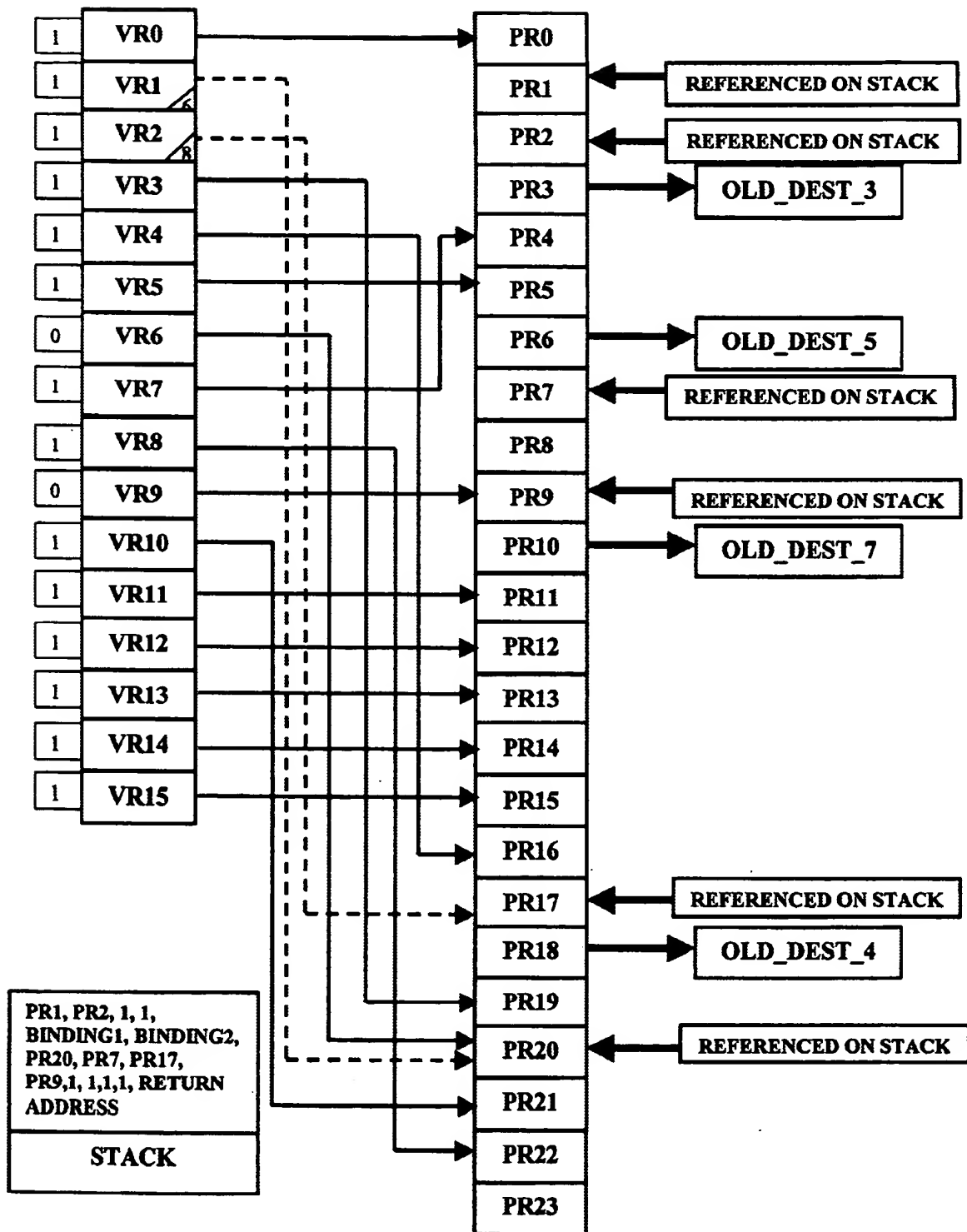
INSTR. 8: SUB VR2, VR3, VR8 maps to SUB PR17, PR19, PR22
1 → DIRTY BIT FOR VR8

FIG. 72

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|------|---|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | - | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | - | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 0 | 1 | 9 | - | WAITING FOR INSTRUCTION 3 TO RETIRE |
| 4 | 1 | - | - | - | INSTRUCTION 1 RETIRED |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | - | WAITING FOR 5 TO RETIRE |
| 7 | 0 | 1 | 17 | 7 | REFERENCED ON STACK |
| 8 | 1 | - | - | - | INSTRUCTION 2 RETIRED |
| 9 | 0 | 1 | 21 | 9 | REFERENCED ON STACK |
| 10 | 0 | 1 | 23 | - | WAITING FOR INSTRUCTION 7 TO RETIRE |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED, |
| 17 | 0 | 1 | 22 | 2 | INS. 2 EXECUTED, REFERENCED ON STACK |
| 18 | 0 | 1 | 6 | - | INS. 3 EXECUTED WAITING FOR 4 TO RETIRE |
| 19 | 0 | 0 | - | 3 | WAITING FOR INSTRUCTION 4 TO EXECUTE |
| 20 | 0 | 0 | - | 6, 1 | WAIT FOR INS. 5 TO EXEC., REF'D. ON STACK |
| 21 | 0 | 0 | - | 10 | WAITING FOR INSTRUCTION 7 TO EXECUTE |
| 22 | 0 | 0 | - | 8 | WAITING FOR INSTRUCTION 8 TO EXECUTE |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 5: DECODE STAGE
INSTRUCTIONS 7 & 8 PHYSICAL REGISTER STATE

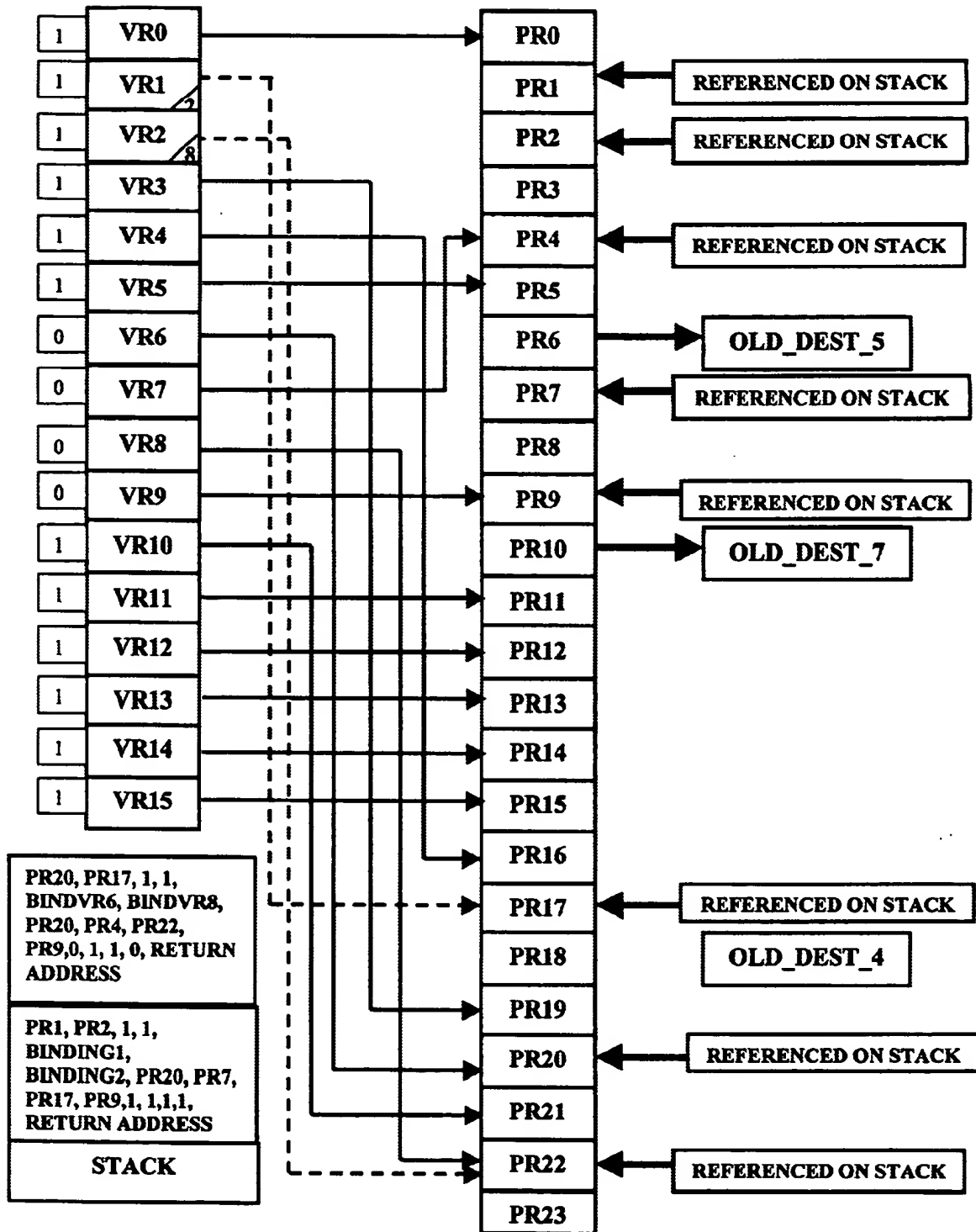
FIG. 73



INSTR. 9: MUL VR8, VR1, VR7 maps to MUL PR22, PR20, PR4
1 → DIRTY BIT FOR VR7

FIG. 74

72/90



INSTR. 10: CALL B,VR2,VR8 action PUSH PR20, PR17, 1, 1, BINDVR6, BINDVR8, PR20, PR4, PR22, PR9, 0, 1, 1, 0, RETURN ADDRESS; BINDVR2_PR17, BINDVR8_PR22, DIRTY BITS FOR VR2&8 → DIRTY BITS FOR VR1&2, 0000 → DIRTY BITS FOR VR6-9, transfer to B

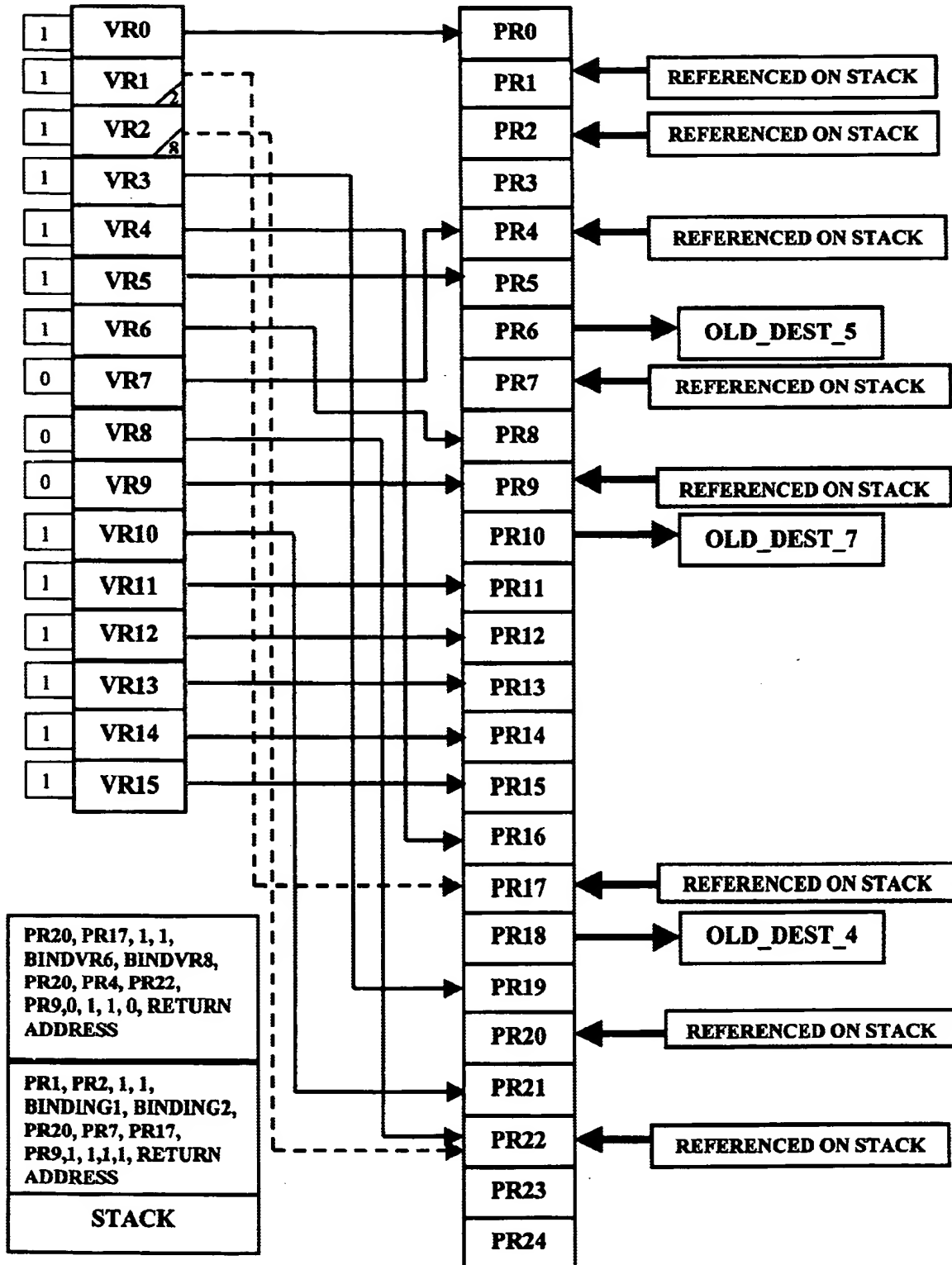
FIG. 75

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|------|---|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | - | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | - | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 1 | - | - | - | INSTRUCTION 3 RETIRED |
| 4 | 0 | 0 | - | 7 | WAIT FOR INS. 9 TO EXEC., REF'D. ON STACK |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | - | WAITING FOR 5 TO RETIRE |
| 7 | 0 | 1 | 17 | - | REFERENCE PREVIOUSLY SAVED ON STACK |
| 8 | 1 | - | - | - | UNALLOCATED |
| 9 | 0 | 1 | 21 | 9 | REFERENCE PREVIOUSLY SAVED ON STACK |
| 10 | 0 | 1 | 23 | - | WAITING FOR INSTRUCTION 7 TO RETIRE |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | -, 1 | INS. 2 EXECUTED, REF'D. ON STACK |
| 18 | 0 | 1 | 6 | - | WAITING FOR INSTRUCTION 4 TO RETIRE |
| 19 | 0 | 0 | - | 3 | WAITING FOR INSTRUCTION 4 TO EXECUTE |
| 20 | 0 | 1 | 78 | 6 | INS. 5 EXECUTED, REF'D. ON STACK |
| 21 | 0 | 0 | - | 10 | WAITING FOR INSTRUCTION 7 TO EXECUTE |
| 22 | 0 | 0 | - | 8, 2 | WAIT FOR INS. 8 TO EXEC., REF'D. ON STACK |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 6: DECODE STAGE
INSTRUCTIONS 9 & 10 PHYSICAL REGISTER STATE

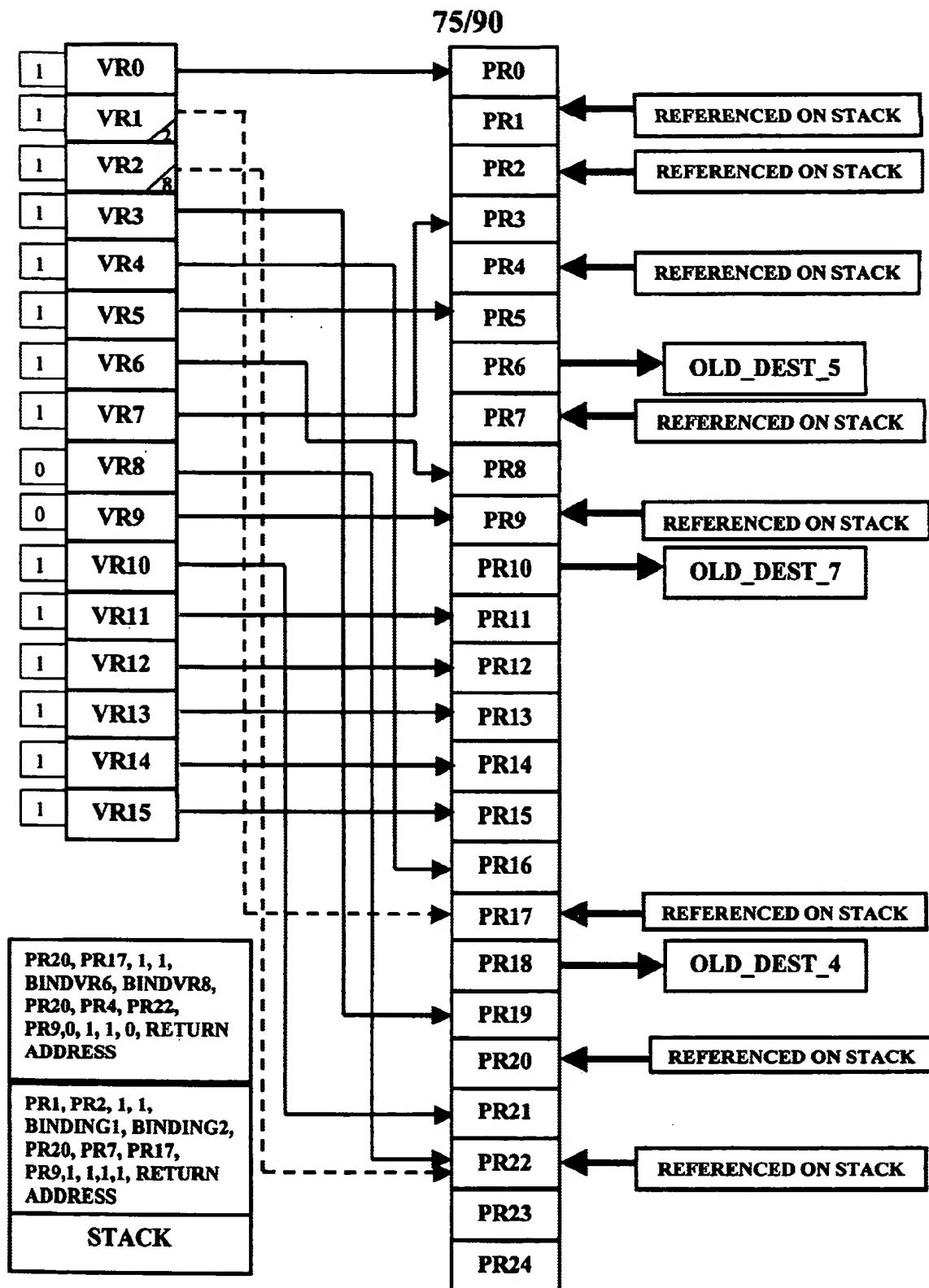
FIG. 76

74/90



INSTR. 11: ADD VR1, VR2, VR6 maps to ADD PR17, PR22, PR8
1 → DIRTY BIT FOR VR6

FIG. 77



INSTR. 12: ADD VR3, VR7, VR7 maps to ADD PR19, PR4, PR3
1 → DIRTY BIT FOR VR7

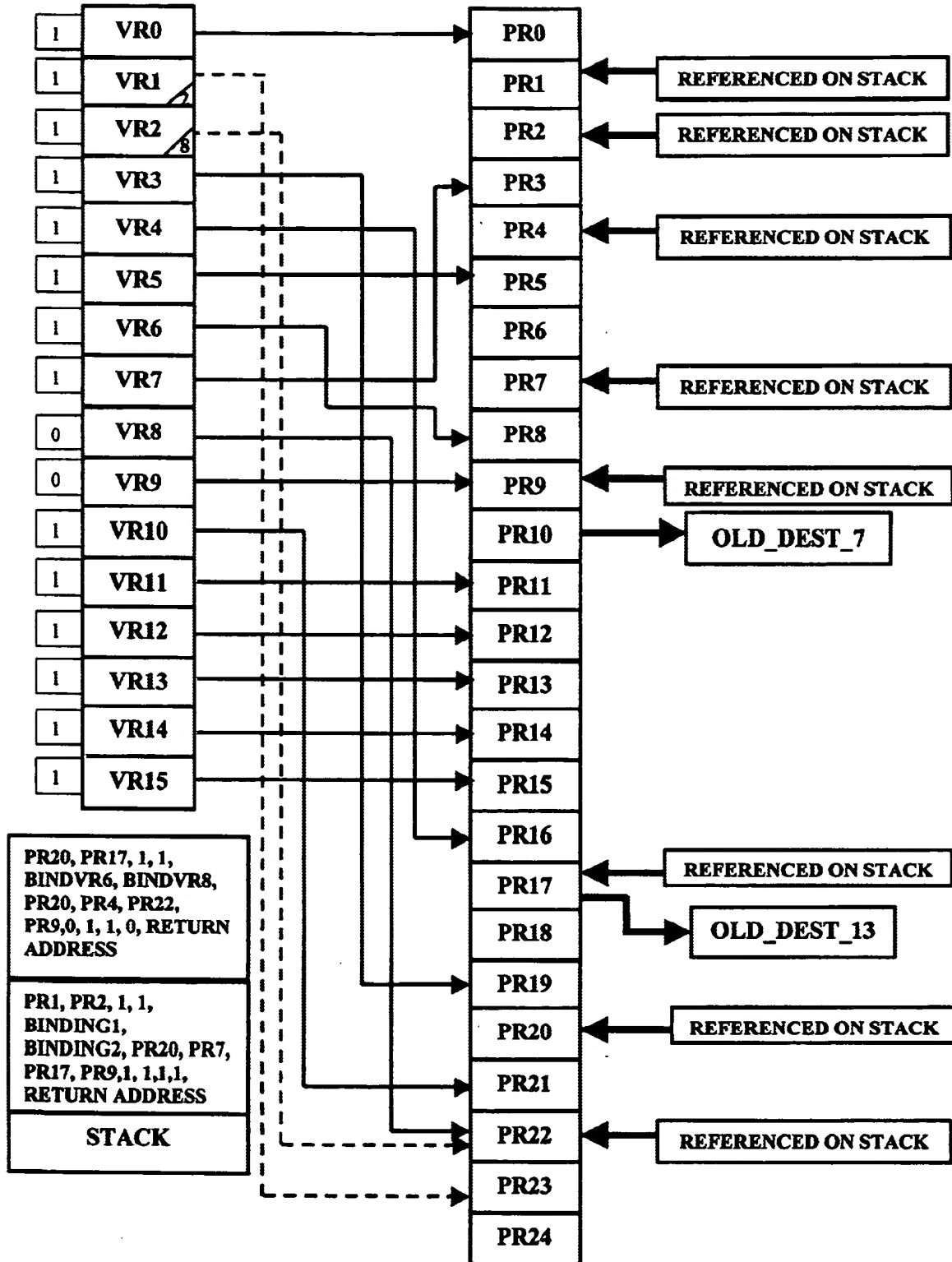
FIG. 78

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-------|---|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | - | REF'D. ON STACK |
| 2 | 0 | 1 | 7 | - | REF'D. ON STACK |
| 3 | 0 | 0 | - | 7 | WAITING FOR INSTRUCTION 12 TO EXECUTE |
| 4 | 0 | 0 | - | - | WAIT FOR INS. 9 TO EXECUTE, REF. SAVED |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 15 | - | WAITING FOR 5 TO RETIRE |
| 7 | 0 | 1 | 17 | - | REF'D. ON STACK |
| 8 | 0 | 0 | - | 6 | WAITING FOR INSTRUCTION 11 TO EXECUTE |
| 9 | 0 | 1 | 21 | 9 | REF'D. ON STACK |
| 10 | 0 | 1 | 23 | - | WAITING FOR INSTRUCTION 7 TO RETIRE |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | - , 1 | INS. 2 EXECUTED, REF'D. ON STACK |
| 18 | 0 | 1 | 6 | - | WAITING FOR INSTRUCTION 4 TO RETIRE |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 78 | - | REF'D. ON STACK |
| 21 | 0 | 0 | - | 10 | WAITING FOR INSTRUCTION 7 TO EXECUTE |
| 22 | 0 | 0 | - | 8, 2 | WAIT FOR INS. 8 TO EXEC., REF'D. ON STACK |
| 23 | 1 | - | - | - | UNALLOCATED |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 7: DECODE STAGE
INSTRUCTIONS 11 & 12 PHYSICAL REGISTER STATE

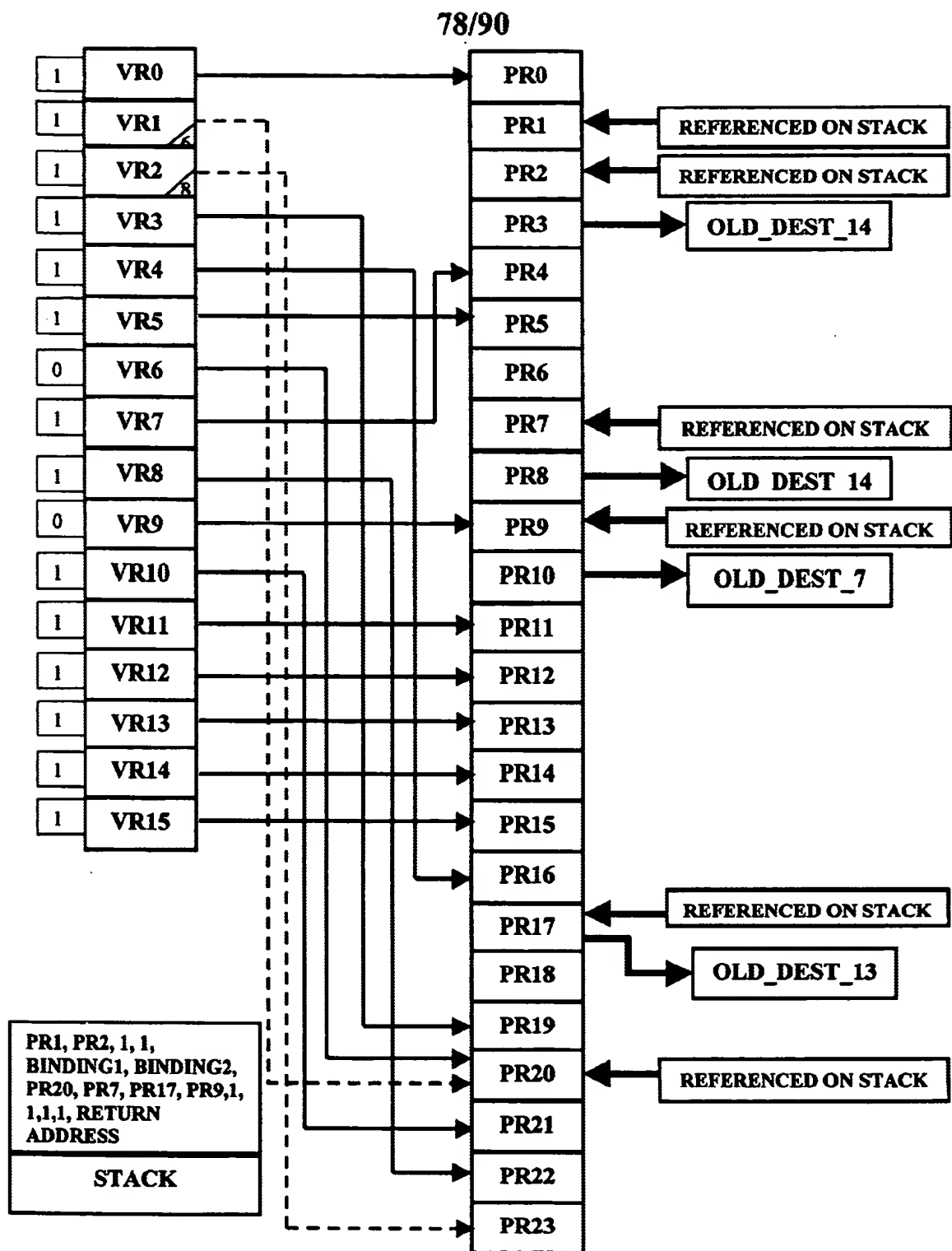
FIG. 79

77/90



INSTR. 13: MUL VR6, VR7, VR1 maps to MUL PR8, PR3, PR23
PR17 → OLD_DEST_13

FIG. 80

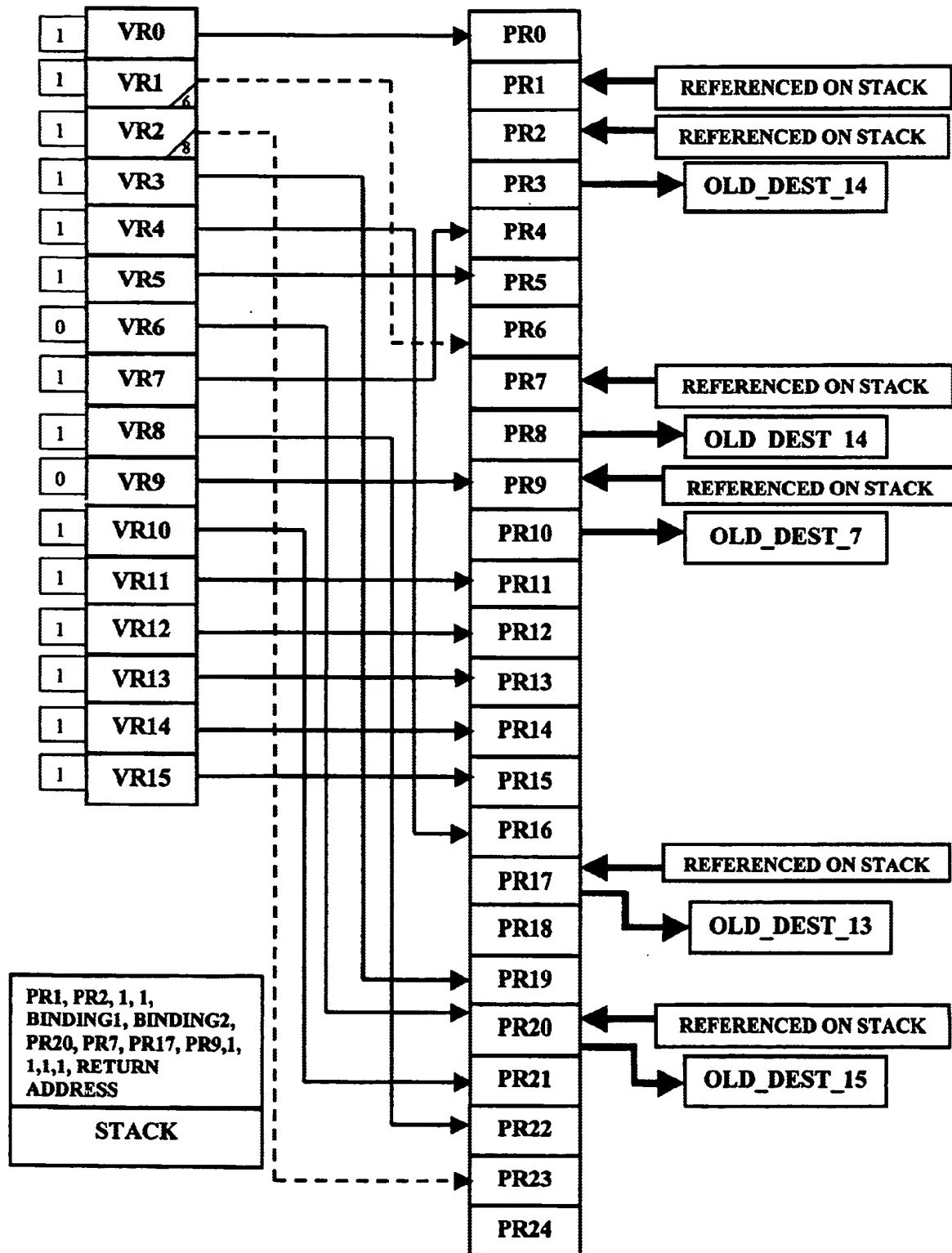


INSTR. 14: RET maps to POP PR20, PR4, PR22, PR9 → VR6-9, 0110 → VR6-9's DIRTY BITS,
 PR20 & PR17 → VR1&2, 11 → DIRTY BITS FOR VR1&2, BINDINGS
 6 AND 8 → BINDINGS FOR VR1 AND VR2, RETURN FROM SUBR. B; OLD VABR1's PR23 → VR2 & OLD VABR1's DIRTY BIT → VR2's DIRTY BIT, OLD VABR2's PR22 → VR8 & OLD VABR2's DIRTY BIT → VR8's DIRTY BIT, PR3 & PR8 → OLD_DEST_14

FIG. 81

CLOCK 8: DECODE STAGE
INSTRUCTIONS 13 & 14 PHYSICAL REGISTER STATE

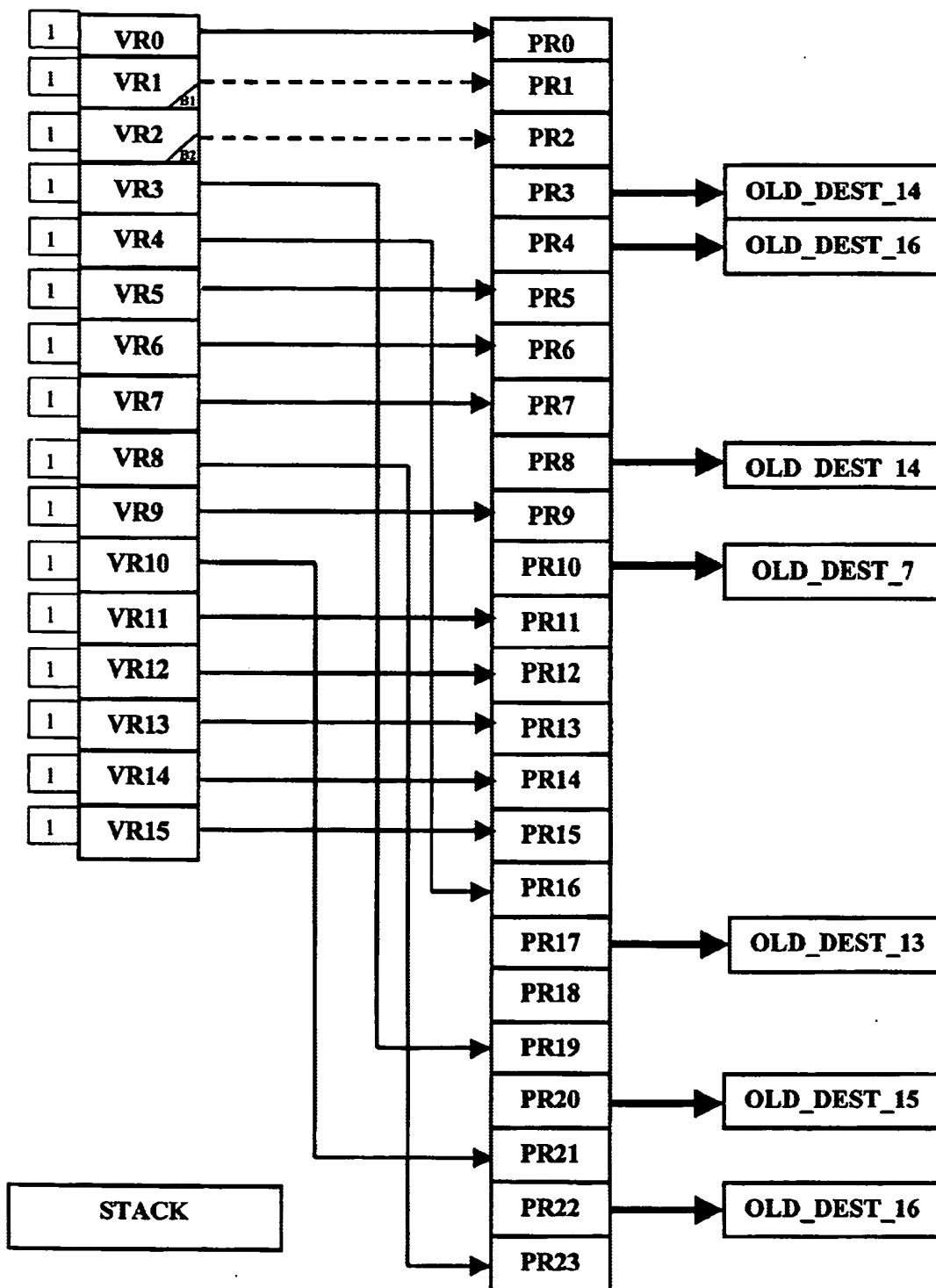
FIG. 82



INSTR. 15: ADD VR8, VR7, VR1 maps to ADD PR22, PR4, PR6
PR23 → OLD_DEST_15

FIG. 83

81/90



INSTR. 16: RET maps to
1111 → VR6-9's DIRTY BITS,

POP PR20, PR7, PR17, PR9 → VR6-9,

PR1 & PR2 → VR1&2, 11 → DIRTY BITS FOR VR1&2, BINDINGS

B1 AND B2 → BINDINGS FOR VR1 AND VR2, RETURN FROM SUBR. A;

OLD VABR1's PR6 → VR6, OLD VABR2's PR23 → VR8,

OLD VABR1&2 DIRTY BITS → DIRTY BITS FOR VR6 & 8, PR4 & PR22 → OLD_DEST_16

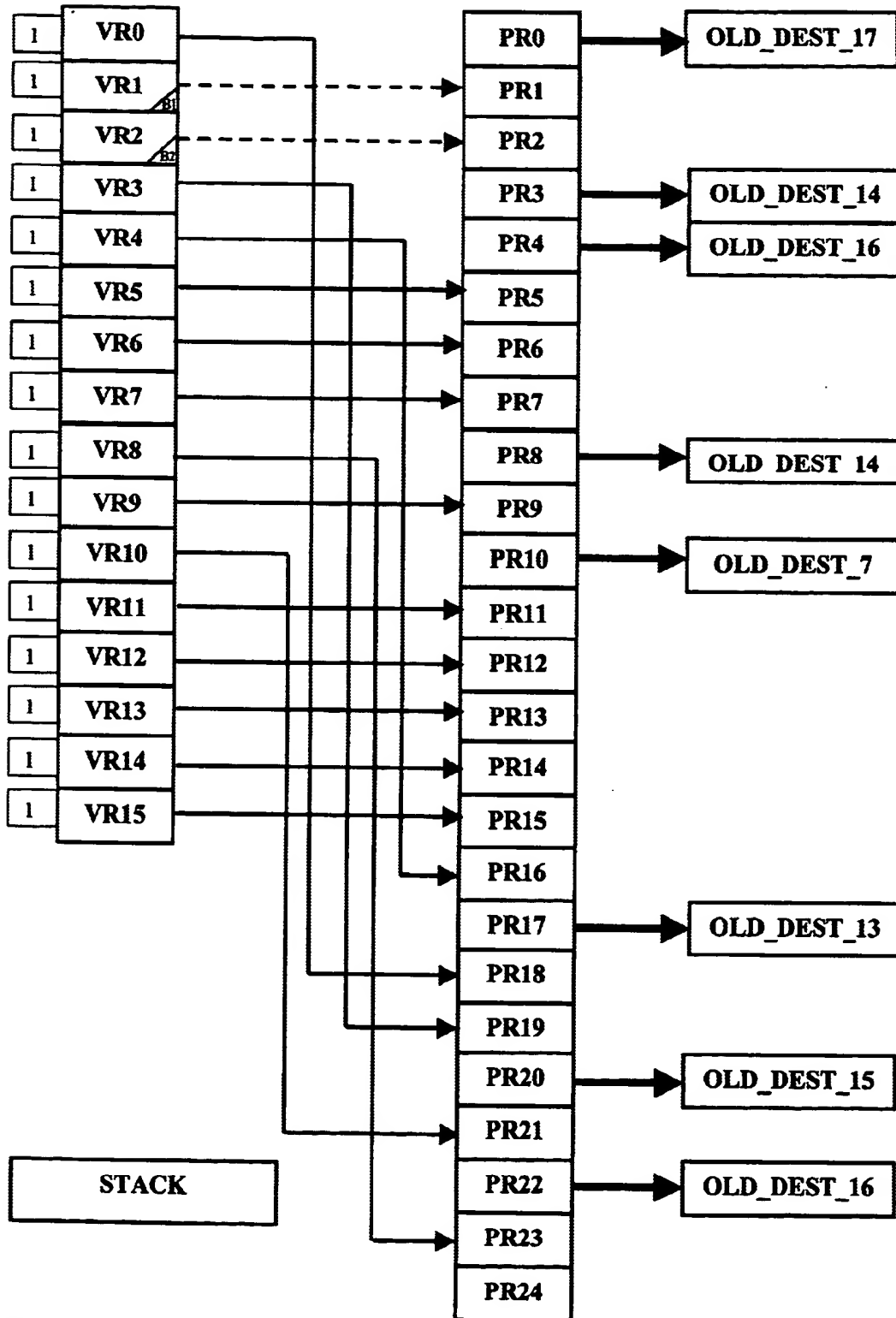
FIG. 84

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|--|
| 0 | 0 | 1 | 3 | 0 | EXAMPLE INITIALIZATION |
| 1 | 0 | 1 | 5 | 1 | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | 2 | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 0 | 0 | - | - | WAIT FOR INS. 12 TO EXEC. & 14 TO RETIRE |
| 4 | 0 | 0 | - | - | WAIT FOR INS. 9 EXEC., INS. 16 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 0 | - | 6 | WAITING FOR INSTRUCTION 15 TO EXECUTE |
| 7 | 0 | 1 | 17 | 7 | VR7 REFERENCE RESTORED FROM STACK |
| 8 | 0 | 0 | - | - | WAIT FOR INS. 11 TO EXEC. & 14 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 0 | 1 | 23 | - | WAITING FOR INSTRUCTION 7 TO RETIRE |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | - | WAITING FOR INS. 13 TO RETIRE |
| 18 | 1 | - | - | - | INSTRUCTION 4 RETIRED, UNALLOCATED |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 78 | - | WAITING FOR INS. 15 TO RETIRE |
| 21 | 0 | 1 | 90 | 10 | INSTRUCTION 7 EXECUTED |
| 22 | 0 | 1 | 10 | - | WAIT FOR INS. 16 TO RETIRE |
| 23 | 0 | 0 | - | 8 | WAIT FOR INS. 13 TO EXEC. |
| 24 | 1 | - | - | - | UNALLOCATED |
| ETC. | 1 | - | - | - | UNALLOCATED |

CLOCK 9: DECODE STAGE
INSTRUCTIONS 15 & 16 PHYSICAL REGISTER STATE

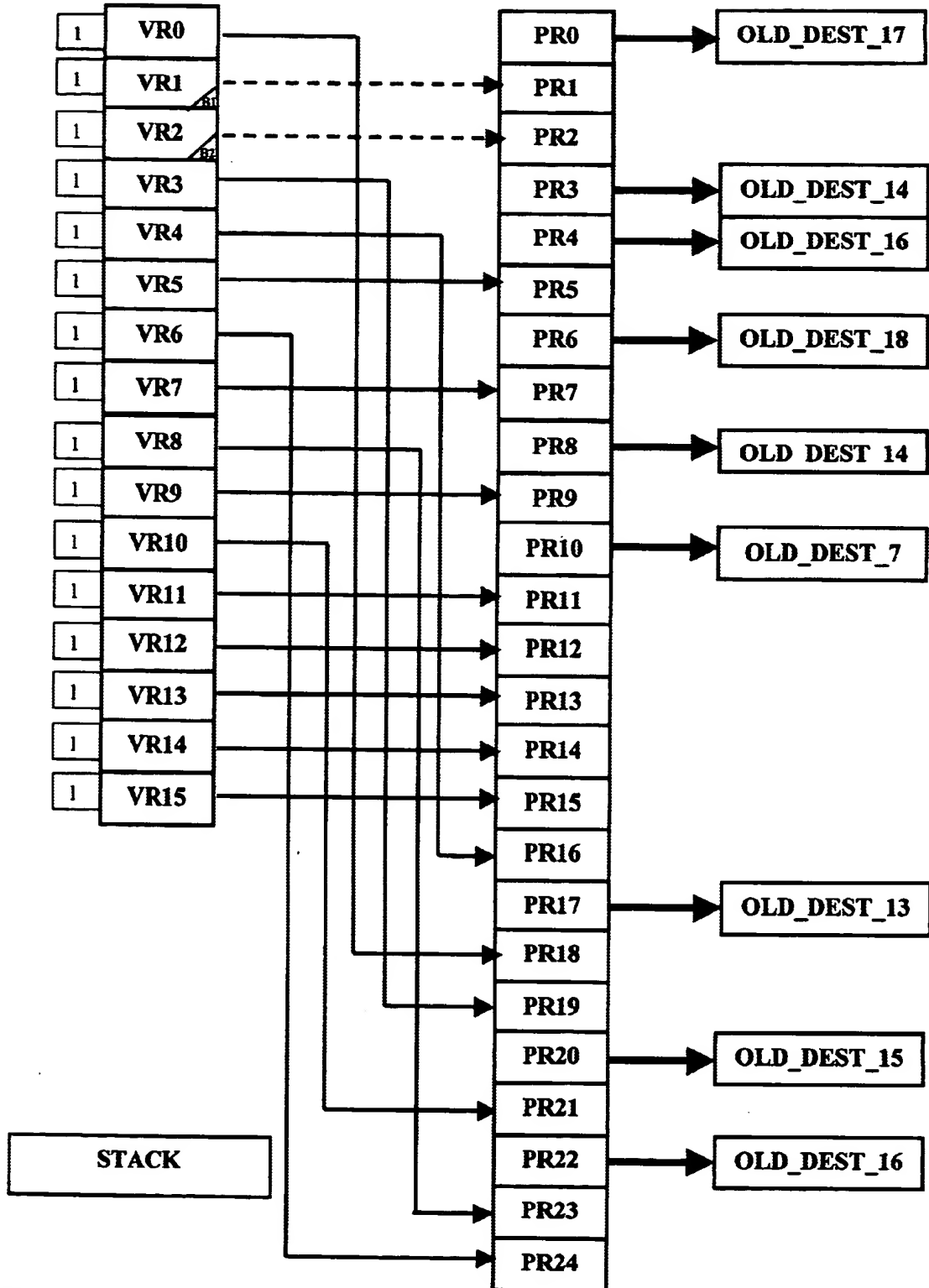
FIG. 85

83/90



INSTR. 17: ADD VR8, VR0, VR0 maps to ADD PR23, PR0, PR18
PR0 → OLD_DEST_17

FIG. 86



INSTR. 18: ADD VR8, VR6, VR6 maps to ADD PR23, PR6, PR24
PR6 → OLD_DEST_18

FIG. 87

CLOCK 10: DECODE STAGE
INSTRUCTIONS 17 & 18 PHYSICAL REGISTER STATE

FIG. 88

CLOCK 11: DECODE STAGE
NO CHANGE IN PHYSICAL REGISTER STATE

FIG. 89

CLOCK 12: DECODE STAGE
PHYSICAL REGISTER STATE

FIG. 90

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|-----------------------------------|
| 0 | 0 | 1 | 3 | - | WAITING FOR INS. 17 TO RETIRE |
| 1 | 0 | 1 | 5 | 1 | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | 2 | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 0 | 1 | 792 | - | WAIT FOR INS. 14 TO RETIRE |
| 4 | 0 | 1 | 780 | - | WAIT FOR INS. 16 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 790 | - | WAIT FOR INS.18 TO RETIRE |
| 7 | 0 | 1 | 17 | 7 | VR7 REFERENCE RESTORED FROM STACK |
| 8 | 0 | 1 | 32 | - | WAIT FOR INS. 14 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 1 | - | - | - | UNALLOCATED |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | - | WAITING FOR INS. 13 TO RETIRE |
| 18 | 0 | 1 | 25347 | 0 | INSTRUCTION 17 EXECUTED |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 78 | - | WAITING FOR INS. 15 TO RETIRE |
| 21 | 0 | 1 | 90 | 10 | INSTRUCTION 7 EXECUTED |
| 22 | 0 | 1 | 10 | - | WAIT FOR INS. 16 TO RETIRE |
| 23 | 0 | 0 | - | 8 | WAIT FOR INS. 13 TO EXEC. |
| 24 | 0 | 1 | 26134 | 6 | INSTRUCTION 18 EXECUTED |
| ETC. | 1 | - | - | - | UNALLOCATED |

**CLOCK 13: DECODE STAGE
PHYSICAL REGISTER STATE**

FIG. 91

| PHYSICAL REGISTER NUMBER | FREE | VALID RESULT | VALUE | VR# | DESCRIPTION |
|--------------------------|------|--------------|-------|-----|-----------------------------------|
| 0 | 0 | 1 | 3 | - | WAITING FOR INS. 17 TO RETIRE |
| 1 | 0 | 1 | 5 | 1 | PREVIOUSLY BOUND TO 'BINDING1' |
| 2 | 0 | 1 | 7 | 2 | PREVIOUSLY BOUND TO 'BINDING2' |
| 3 | 0 | 1 | 792 | - | WAIT FOR INS. 14 TO RETIRE |
| 4 | 0 | 1 | 780 | - | WAIT FOR INS. 16 TO RETIRE |
| 5 | 0 | 1 | 13 | 5 | EXAMPLE INITIALIZATION |
| 6 | 0 | 1 | 790 | - | WAIT FOR INS.18 TO RETIRE |
| 7 | 0 | 1 | 17 | 7 | VR7 REFERENCE RESTORED FROM STACK |
| 8 | 0 | 1 | 32 | - | WAIT FOR INS. 14 TO RETIRE |
| 9 | 0 | 1 | 21 | 9 | EXAMPLE INITIALIZATION |
| 10 | 1 | - | - | - | UNALLOCATED |
| 11 | 0 | 1 | 25 | 11 | EXAMPLE INITIALIZATION |
| 12 | 0 | 1 | 27 | 12 | EXAMPLE INITIALIZATION |
| 13 | 0 | 1 | 29 | 13 | EXAMPLE INITIALIZATION |
| 14 | 0 | 1 | 31 | 14 | EXAMPLE INITIALIZATION |
| 15 | 0 | 1 | 33 | 15 | EXAMPLE INITIALIZATION |
| 16 | 0 | 1 | 6 | 4 | INSTRUCTION 1 EXECUTED |
| 17 | 0 | 1 | 22 | - | WAITING FOR INS. 13 TO RETIRE |
| 18 | 0 | 1 | 25347 | 0 | INSTRUCTION 17 EXECUTED |
| 19 | 0 | 1 | 12 | 3 | INSTRUCTION 4 EXECUTED |
| 20 | 0 | 1 | 78 | - | WAITING FOR INS. 15 TO RETIRE |
| 21 | 0 | 1 | 90 | 10 | INSTRUCTION 7 EXECUTED |
| 22 | 0 | 1 | 10 | - | WAIT FOR INS. 16 TO RETIRE |
| 23 | 0 | 1 | 25344 | 8 | INSTRUCTION 13 EXECUTED |
| 24 | 0 | 1 | 26134 | 6 | INSTRUCTION 18 EXECUTED |
| ETC. | 1 | - | - | - | UNALLOCATED |

**CLOCK 14: DECODE STAGE
PHYSICAL REGISTER STATE**

FIG. 92

CLOCK 15: DECODE STAGE
PHYSICAL REGISTER STATE

FIG. 93

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☒ **OTHER:** Copied documents

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.